快速更新循环同化预报系统 Documentation

发布 *0*

华智睿选（武汉）科技有限公司

2020 年 04 月 23 日

Contents:

CHAPTER 1

系统简介

1.1 快速更新同化预报系统简介

自 1994 年业务化以来, 美国 NCEP(National Centers of Environmental Prediction) 的快速更新同化预报系统 (Rapid Update Cycle, 简称 RUC) 所提供的 0~12h 预报已成为航空气象、强对流灾害天气等主要的中尺度预报指导产品。RUC 由同化和预报两个部分构成, 通过高频次的资料同化吸收最新的观测信息, 不断更新模式背景场, 形成更准确的初始场并进行短期预报。自业务运行以来,RUC 系统进行了多次改进与升级, 新一代的快速更新同化预报系统 (Repaid Refresh, 简称 RR) 于 2012 年正式业务化并取代原来的 RUC 系统 (http://repaidrefresh.noaa.gov)。

针对强对流天气系统的 0~6h 预报, 各大气象研究和业务中心主要基于雷达回波与卫星图像的简单外推以及实践经验, 发展了各种类型的短时临近预报 (NOWCASTING) 系统, 业务应用表明它们在 0~1h 的预报中相当有效。然而由于缺乏对强对流系统的发生、发展和消亡的物理机制描述, 其预报能力随预报时效增加迅速降低。如下图中的蓝线所示, 一般而言时效超过 1h 以上的预报可信度大大降低, 尤其是对强对流系统发展、演变的预报。采用数值模式预报强对流系统, 虽然对动力与物理过程的描述存在着各种各样的不足, 但对强对流系统活动的预报在原理上应该远优于简单的外推方法。鉴于强对流系统水平尺度较小、生命史较短的特点, 模式初始时刻对当前对流系统的准确把握是关键。传统的每 6h 或更长时间更新一次模式初值的数值预报系统显然不适合于短临时间尺度的预报, 因此进行较短时间间隔的高频资料同化以便初始场尽可能包含对流系统的信息就显得十分必要。美国在 1991 年开始研究快速更新同化技术, 并于 1994 年开始业务化运行。但是高频资料同化存在着模式初始变量不协调而引发的较长时间 (>1h) 的初始平衡问题与误差积累等问题, 它们可以大大削弱模式 1~2h 的预报能力, 为此所谓的热启动技术就应运而生。需要指出的是, 对于强对流天气系统的短时临近预报, 基于快速更新同化的数值预报系统与基于外推和经验的短临系统相结合 (blending) 是提高短时临近预报准确率的重要方向 [chenbaode2013]。

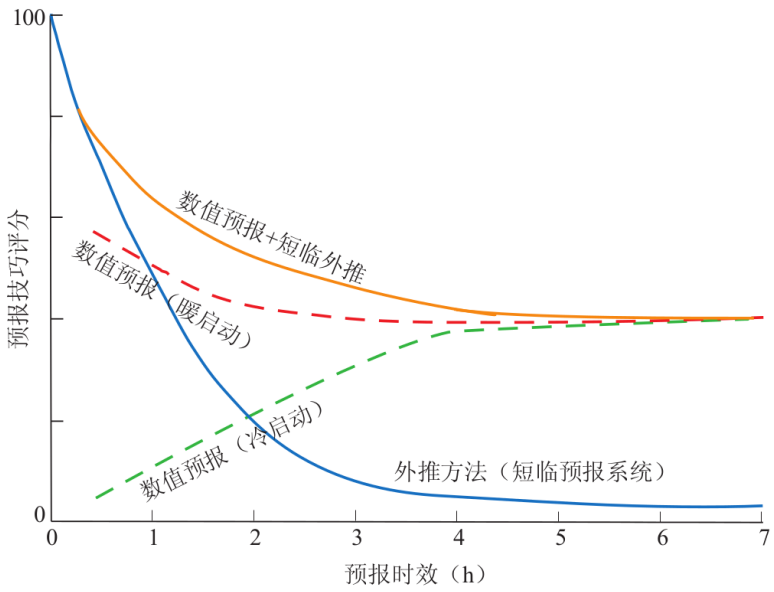


图 1: 数值预报技术、短时临近预报技术预报技巧比较示意图

1.2 发展历史

目录

1.2.1 美国 RUC 发展历史

美国首先在 1991 年开始了快速更新同化预报技术的研究开发, 并于 1994 年开始在 NCEP 业务运行 (RUC1)。RUC1 采用静力平衡原始方程模式, 水平分辨率 60km, 垂直方向 25 层, 使用最优插值 (Optimal Interpolation, OI) 技术每 3h 进行一次资料同化更新; 之后经过几次更新和升级 2005 年系统水平分辨率已提升至 13km, 垂直方向 50 层, 采用 3 维变分 (3DVAR) 技术, 进行 1h 同化更新。为得到更准确的云和水物质初始场, 从 2002 年 (RUC20) 起, 系统开始使用云分析技术利用卫星观, 之后通过逐步补充更多的观测资料 (包括雷达反射率), 对云的描述更加完善。云分析技术在 RUC 系统初始化过程中起着至关重要的作用。

虽然 RUC 系统经历了多次升级, 但其模式动力框架还是静力模式。随着非静力模式的发展, NOAA/NCEP(RUC 系统业务应用部门) 和 NOAA/ESRL/GSD(RUC 系统研发机构) 于 2002 年决定发展以非静力模式 WRF 为基础的新一代快速更新同化预报系统, 除模式替换外, 同化系统也将换成 GSI(GridpointStatistical Interpolation) 以增加系统对卫星辐射率的直接同化功能。自 2003 年以来, 在经过了 WRF-RUC 测试 (以 RUC 初始场驱动 WRF, Weather Researchand Forecasting)、WRF 动力框架比较

测试 (在 WRF-ARW 和 WRF-NMM 两种动力框架中选择了 WRF-ARW)、GSI 系统功能拓展 (移植 RUC 已有的云分析、雷达反射率资料同化、非绝热 DFI、地表资料同化等技术)、云分析模块升级 (增加深对流云量条件下, 连续高频次的低密度资料同化效果可能劣于低频次的高密度资料同化。这种时、空分辨率上的两难取舍取决于风暴的移速, 当风暴准静止时, 同化频率可取的非常低以获得空间上高密度的资料; 当风暴移速较快时, 要适当增加同化频率以降低时间代表性误差, 但太高的频率又会造成资料空间代表性上的不足, 影响同化效果。

1.2.2 中国 RUC 发展历史

近几年, 我国加大了针对中小尺度天气系统观测的布网力度, 观测手段及自动化水平逐步接近国际先进水平, 具备了发展快速同化系统的观测基础。在我国东部地区已建成包括自动气象站、单双频地基 GPS(Global Position System) 站、风廓线仪、多普勒雷达和微波辐射计在内的局地中小尺度观测网, 观测资料在时间和空间分辨率上都得到了极大的提高。通常一个时次的常规地面观测超过 1000 个, 自动站资料可达 2 万个以上, 雷达等非常规探测在硬件建设和布网上发展非常迅速。目前, 长三角地区已有 13 部多普勒天气雷达 (徐州、连云港、盐城、南京、南通、上海南汇及青浦、杭州、金华、宁波、舟山、温州、衢州), 在建 4 部 (江苏常州、浙江湖州、杭州及丽水)。计划建设 25 部风廓线仪, 其中江苏新建 6 部, 浙江新建 5 部 (已建成湖州、义乌、嘉兴), 上海新建 11 部 (已建成 10 部), 并完成长三角风廓线组网。长三角区域将有 79 个 GPS 站可参与加密观测, 其中参与中国气象局业务考核的有江苏 62 个、上海 9 个、浙江 1 个。这些观测在日常业务预报与科学研究中起着举足轻重的作用。如何有效利用各种常规和非常规气象资料, 充分发挥先进探测手段的作用, 提高突发灾害性天气的预报能力, 提供时间上更密集、空间和量级上更准确的精细数值预报服务, 已成为日益突出的问题。在国内, 虽然对快速更新同化和预报技术的研究和应用起步较晚, 但近年来发展迅速。以上海、北京及广州为代表, 基于本地特色, 分别建立了不同技术路线的快速更新同化预报系统。

2009 年 1 月, 上海建立了基于 ADAS 资料同化系统和区域中尺度数值模式 WRF 的快速更新同化预报系统 (SMB-WARR, Shanghai Meteorological Bureau-WRF ADAS Rapid Refresh System), 同年 7 月准业务运行, 2009 年底实现业务运行至今。该系统的水平分辨率为 3km, 垂直分辨率为 51 层, 预报区域覆盖华东及其周边地区, 系统采用逐小时循环同化的方式对观测资料进行同化, 每日 02 时冷启动, 冷启动的背景场和侧边界条件由 NCEP GFS 6h 的全球预报场提供, 其他时刻同化的背景场则由 SMB-WARR 系统上一时次的 1h 预报场提供。系统每小时启动一次预报, 预报时效为 12h, 并提供逐小时高分辨率中尺度分析场。实时业务的同化观测资料包括常规天气观测、机场地面报、船舶、浮标、自动站、飞机报、探空、雷达反射率和 FY-2E 红外和可见光辐射率资料, 如图 2 所示。其中, 雷达反射率和 FY-2E 红外和可见光辐射率资料主要采用云分析技术进行同化, 雷达反射率资料主要用于改进和修正云和降水的垂直结构特征, 而 FY-2E 红外和可见光辐射率观测则主要用于修正初始场中有关云顶高度和云量的特征, 关于云分析技术的具体叙述。其他观测资料则是通过 ADAS 的逐步订正方法实现同化。通过 3 年来的测试及优化, 该系统在短时临近天气预报中发挥了越来越重要的作用; 上海地区 2011 年汛期强对流预报检验表明, 其对午后强对流天气的预报明显优于同时次起报的华东区域中尺度预报模式 (SMB-WARMS) 和欧洲中心的高分辨模式的预报结果。

北京同化预报系统 (BJ-RUC) 亦采用 WRF 模式, 与 SMB-WARR 不同之处在于其同化系统采用三维变分技术。区域配置采用三重嵌套 27×9×3km, 3km 分辨率的预报区域主要覆盖北京及其周边地区, 垂直分辨率均为 37 层。每日 20 时冷启动, 冷启动的背景场和侧边界条件由 NCEP GFS 6h 的全球预报场提供, 每隔 3h 同化一次探空、地面、船舶观测资料以及北京地区的自动站和地基 GPS PW 可降水量观测资料, 预报时效为 18h。其业务运行结果表明, 3km 分辨率无论是降水时段、落区和雨量均较 9km 分辨率有更好的预报效果, 尤其是大量级降水的预报, 但系统对局地对流降水的预报能力仍然有限。

广州在 GRAPES_meso 及三维变分同化基础上发展了快速更新同化预报系统 (GRAPES-CHAF)。模式分辨率为 0.12° , 预报区域主要覆盖华南地区。采用三维变分同化方法对卫星反演的云迹风、雷达径向风、VAD 风、飞机报、地面、船舶和探空观测资料进行同化, 通过云分析技术和 nudging 方法对雷达回波进行同化。该系统采用逐小时循环同化和每 3h 间隔的滚动预报, 预报时效为 24h。多种测试和汛期连续试验表明, 系统运行稳定可靠。通过 1 个月的滚动预报综合分析和个例预报分析均表明, 预报稳定有效, 与观测分布基本一致, 初步具备开展短时临近预报的能力。在原广州热带海洋气象研究所 GRAPES-CHAF 基础上, 国家气象中心采用新版的 GRAPES-Meso 模式与区域 GRAPES-3DVAR, 建立了优化后的 RUC 同化预报系统。其模式分辨率为 0.15°, 预报区域覆盖全国, 其同化的观测资料类型与 GRAPES-CHAF 类似。每 3h 启动一次预报, 预报时效为 24h。

对比国内各家快速更新同化预报系统, 北京和广州均采用 3DVAR 而上海使用 ADAS 方案; 从同化资料的种类看, 广州的系统资料种类相对较多 (如云导风、雷达径向风等)。然而和美国的 RUC/RR 系统相比, 国内系统可同化的高空资料种类和数量严重不足, 尤其是飞机报资料。飞机报是美国 RUC/RR 系统高频资料的重要来源, 每小时的资料量为数千个, 而国内使用的飞机报不到上述资料量的十分之一。此外, 国内卫星、雷达及其反演资料的质量也一定程度上影响了快速更新同化预报系统的发展。

1.3 关键技术

目录

1.3.1 绝热与非绝热数字初始化

由于模式本身和初始场均不是完全准确的, 模式预报不可避免地存在误差, 且预报误差随预报时效非线性增长, 资料同化的目的即是利用观测资料修正模式预报。一般天气尺度模式系统同化的频率是每 6h 一次, 而对流系统因其发展迅猛、生命史短、预报误差增长快等特点, 需要高频次观测资料对模式初值不断同化更新, 以得到更精准的模式初值, 继而提高预报准确率, 因此高频资料同化对强对流系统预报的好处是显而易见的。

然而, 高频资料同化也会带来一些问题。由于观测类型的不完整性及资料本身的误差, 同化后的变量之间不可避免地存在物理不平衡性, 由此激发的重力波将会导致预报误差的异常增长, 这些虚假重力波的影响在模式积分一段时间后会通过平衡调整逐渐消失, 一般将此过程称为平衡调整过程 (spin-up), 所需时间称为 spin-up

时间。模式的预报只有在调整期后才是可信的。然而, 高频同化会不断累加、放大这种重力波的噪音效应, 使预报结果甚至差于一般同化频率的模式预报。为控制重力波噪音, 在高频同化系统中需要使用数字滤波初始化技术 (digital filter initialization, DFI) 来过滤掉初始场中的高频噪音 (重力波)。早期的系统使用绝热的 DFI 即在模式预报之前对初值场进行向前和向后一段时间的绝热积分。近年来快速更新同化预报系统开始使用基于雷达反射率的非绝热 DFI 技术 (diabatic digital filter initialization, DDFI)。DDFI 与 DFI 的差别在于 DDFI 在向后一段时间积分时, 引入了由雷达反射率通过云分析所确定的潜热加热率。它的主要作用在于其可以修正初始风场, 以使得风场与初始对流活动大体上相匹配。

1.3.2 冷启动、暖启动和热启动技术与循环同化设计

尽量缩短 spin-up 时间是提高快速更新同化预报系统性能和效率的关键之一, 模式的冷启动 (Cold Start)、暖启动 (Warm Start) 和热启动 (Hot Start) 技术与循环同化方案中对各种启动技术的合理配置, 都是针对如何尽量缩短 spin-up 时间的。模式冷启动指区域模式背景场由其他低分辨模式 (如全球模式) 提供, 以此为背景场做同化或直接当作初始分析场进行预报, 它的好处是可以保证模式预报不偏离大尺度背景或环境场, 但由于初始场中缺乏中尺度信息, 特别是云和水物质的特征量一般而言为零, 模式需要较长的 spin-up 时间。模式暖启动与热启动时, 同化背景场来自系统模式自身上一时刻的预报, 这时的背景场已包括云和水物质的特征量, 如果在同化时不进行云和水物质的调整 (通常初始垂直速度也取为零), 直接进入模式的下次预报, 称为暖启动; 如果在同化时通过雷达资料直接同化或使用复杂云分析对与对流系统相关的云和水物质也进行调整, 并且也调整相应的垂直速度, 使用这种初始场启动模式, 称为热启动。将各种启动方式按时间合理组合, 进行连续各时次预报就构成了循环同化预报。这里需要指出循环同化预报技术对中尺度天气的短时临近预报十分必要, 因其:1) 允许高分辨模式保留模式上一循环中所发展的小尺度特征, 特别是当前观测中没有的相关垂直运动;2) 维持初始场与模式物理参数化的平衡, 例如, 如果初始相对湿度与云参数化方案需要的初始云场不匹配, 模式前几小时的积分会生成过多的云和降水或者使云和降水消失。但是循环同化预报时, 高分辨预报模式会产生虚假的小尺度特征, 由于观测网通常无法解析这些小尺度特征, 也就无法在同化中使用观测来订正。这些误差有时将会被一直代入下一循环的预报, 最终使得预报大幅度偏离大尺度环境场而导致预报完全失败。为保证模式预报不偏离大尺度背景或环境场, 一段时间后模式必须进行冷启动, 而预报系统何时或多长时间进行冷启动, 依赖于实际应用经验和对大气系统的理解, 为体现完整的日变化特征, 一个冷启动到下个冷启动通常取为 24h。考虑到通常夜间对流活动不活跃, 边界层相对稳定, 其选择当地的凌晨 2 时为冷启动时间, 每 24h 系统冷启动一次。

1.3.3 云分析技术

在快速更新同化预报系统中, 每次模式启动时, 需要对模式初始变量 (包括云、水物质含量和地表变量等) 进行更新, 使模式不断吸收反映天气系统时间演变的观测信息从而达到使模式初始场更加接近实际天气状况的目的。要快速获得有关对流的信息 (如云和降水), 并同化进入模式初始场中, 云分析方法 (cloud analysis) 是个不错的选择。国外目前在业务中使用的云分析系统包括美国空军的 CDFS-II(Cloud Depiction and Forecast System), 英国气象局的 Nimrod(Nowcasting and Initialization for Modeling Using Regional Observation Data Scheme), 美国俄克拉荷马大学 CAPS(Center for Analysis and Predictionof Storms) 的 ARPS, NOAA 的 LAPS(Local Analysisand Prediction System) 和 RUC/RR(Rapid UpdateCycling/Rapid Refresh)。云分析方法的基本原理大体上一致, 即首先根据地面、雷达和卫星等观测资料所提供的有关云与降水信息, 结合数值预报模式结果得到云的几何特征 (如云顶和云底高度, 云类型等), 然后通过对云内的温度和水汽场进行

调整 (如基于绝热上升的热力调整) 得到云水物质参数的混合率和分布。以 ARPS 中 ADAS 云分析为例, 它首先通过地面、雷达和卫星等观测资料得到模式格点上的云量和云顶高度; 然后根据三维雷达反射率资料通过 Smith-Feddes 模型, 并考虑干空气夹卷过程及凝结过程对云水量的损耗, 得到云水、云冰混合比; 而雨水、雪、冰雹等降水粒子场则是由模式格点上的温度场和雷达反射率通过雷达反射率方程诊断分析获得; 最后根据与云水、云冰对应的潜热释放, 基于湿绝热假定对云内温度场进行调整。云分析通过综合使用卫星、雷达和地面观测资料逐步完善和实现了对模式初始场的三维云特征量场的构造和修正, 同时现在的云分析方案在对云的云水、云冰特征量的分析时还分别考虑了层云与积云的不同特征, 在物理机制上的设计也更为合理。研究表明使用该方法可以缩短模式 spin-up 时间和提高模式中小尺度对流降水的预报效果。但是需要指出的是, 通过云分析方法得到的云水物质的混合比与模式的初始预报量 (温度、湿度或者垂直速度) 不平衡, 有时会造成模式的积分不稳定等问题, 这些可以通过非绝热 DFI 技术来解决。

1.3.4 近地面资料同化技术

近地面资料由于有较高的时空分辨率 (如每小时的地面自动站 2m 温度和 10m 风观测), 它自然成为快速更新同化预报系统中, 对背景场进行快速更新同化时重要的观测资料来源。尽管各种近地面观测可以较容易得到, 但是实际上将其同化入系统并不容易, 主要原因如下:

1. 缺乏好的或合理的方法将近地面观测与大气的三维结构相关联, 因此近地面观测的作用在模式积分过程中很快就会消失;
2. 如果不进行严格的质量控制和大量的试验, 近地面观测的引入可能会破坏模式预报的边界层结构;
3. 模式地形与观测站的地形高度不一致, 可能对近地面变量特别是气温和风产生严重影响;
4. 近地面观测在某些地区分布太密集以至于观测资料的水平分辨率大于模式的分辨率, 这时可能产生模式和观测的代表性差异, 且过密的资料分布有时可能使分析太接近于观测, 如果观测质量不高且又低估了观测误差时会使得分析精度下降;
5. 同化系统中对近地面资料的观测误差取值有待改进;6) 同化方案中的观测算子 (如基于相似理论) 尚存在较大误差。

有许多研究工作致力于解决上述问题, 如美国的 RUC 系统利用模式背景场提供的局地温度递减率将近地面观测资料外插到模式地表高度, 其中外插精度很大程度上取决于背景场对近地层热力稳定度的把握程度。

1.3.5 同化频率与资料空间分辨率

快速更新同化预报系统初值的准确性还取决于资料的空间密度、资料质量及同化方法。时间上过高的同化频率会导致资料在空间分布上的不足, 继而可能对那些时空分辨率不足的观测资料同化造成负面影响。以飞机报资料为例, 其时空分辨率取决于非常不规则的飞行线路。如果同化间隔为 6h, 则正负 3h 同化窗口内的资料空间分辨率较高, 但存在的问题是窗口内很多资料的观测时间与实际的同化时间并不一致, 对快速移动的风暴系统而言这种时间代表性误差会导致同化的失败; 而当同化间隔缩小至 1h, 正负半小时同化窗口内的资料时间差问题减弱, 辨率也大大降低。Benjamin 等 [1] 指出, 在相同的资料量条件下, 连续高频次的低密度资料同化效果可能劣于低频次的高密度资料同化。这种时、空分辨率上的两难取舍取决于风暴的移速, 当风暴准静止时, 同化频率可取的非常低以获得空间上高密度的资料; 当风暴移速较快时, 要适当增加同化频率以降低时间代表性误差, 但太高的频率又会造成资料空间代表性上的不足, 影响同化效果。

1.3.6 混合数据同化技术

基于美国国家环境预报中心 (NCEP) 的业务快速同化循环更新预报系统 (Rapid Refresh, 即 RAP) 以及业务高分辨率快速同化循环更新预报系统 (High-Resolution Rapid Refresh，即 HRRR), 针对福建以及海峡地区的天气，气候和下垫面特征，结合福建的观测资料和高性能计算资源，我们设计出一套福建高分辨率快速同化循环更新预报系统（FJ-RAP 和 FJ-HRRR）以满足福建短临精细化预报的服务要求。该系统采用的是目前国际上流行的混合数据同化方案，使用的是美国 NCEP 的业务数据同化模式 GSI。

区别于传统的数据同化技术，混合同化是将变分同化和集合卡尔曼滤波有机的耦合在一起，实现他们相互间的优势互补与局限性减弱，从而可以最优化地和最大限度地使用各种现有常规和非常规气象观测数据 (地面, 探空, 自动站, 风廓线, 多普勒雷达, 地基 GPS-MET 和卫星等). 针对不同观测数据，混合同化技术根据实时大气环流信息以及各种物理平衡条件，为短临天气预报模型提供一个最优化的初始条件，从而进一步提升模式预报结果的准确度。另外，根据观测数据的实时传输情况，混合同化技术可以将最新的数据引入预报模式，从而提供不断更新的短期临近预报结果。

对于一般的变分同化而言，其可以描述为对目标泛函 J 进行收敛，从而获得对天气预报的初试场 x 的最优分析。目标泛函 J 的增量（incremental）表述形式为:

以上方程可以转换为控制变量表述的形式, 使用:

同理，另外一种优化方法”集合卡曼滤波”(EnKF), 其本质上和变分方法具有一致性, 只是方式不同, EnKF 使用了一个预报集合对当前大气环流场的不确定性进行显性的估计, 从而建立对背景场误差协方差矩阵的近似分析, 从而对收敛函数的下降方向给出直接迭代关系:

混合数据同化实际上是将变分方法和 EnKF 进行结合，汲取两者之长，从而解决它们自身的弱点。简而言之, 我们可以理解为将 EnKF 中的背景场误差协方差矩阵引入到目标泛函的收敛过程中，从而加入集合预报当前环流场信息到优化方程的收敛过程中:

其中 Pf 就来源于 EnKF. 然而, 为了可以进行计算和迭代求解，我们需要对以上方程进行改写, 并引入转换项 (Alpha-Control Transform):

这样一来经过 Schur 项的背景场误差协方差矩阵局地化空间分析 (Covariance Localization), 目标泛函的收敛方程及变为:

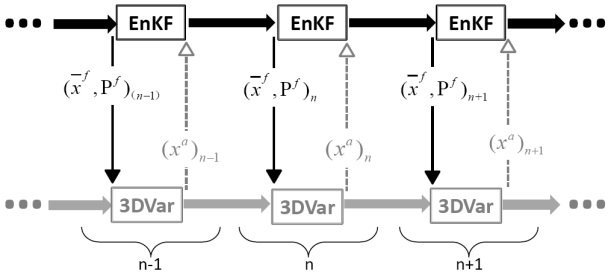


图 2: EnKF-3DVar Hybrid 混合数据同化示意图

本项目混合同化部分, 将选用美国 NCEP 的业务数据同化模式 GSI, 主要考虑的是以下几个因素:

1. GSI 是美国海洋大气局 NOAA 当前稳定运行的先进业务模式，性能稳定，效果经得起考验;

2. 在业务环境下经过多年的磨练，GSI 本身对观测资料也有很完善的质量控制;

3. GSI 功能比较齐全，对几乎所有观测类型都可以同化;

4. GSI 的混合数据同化已经在全球预报（GFS）和区域高分辨率模式中（RAP 和 HRRR）业务化运行;

CHAPTER 2

安装手册

2.1 Implementation

参考文档 Implementation_Standards

2.2 环境变量设置

在 $HOME/.profile, 或 $HOME/.bashrc, 或 $HOME/.cshrc 中加入以下设置:

```
# Define the archor/top directory
# -------------------------------

$ setenv ROOT_DIR /glade/work/xinzhang/repository

# Define common libs
# ------------------------

$ setenv NETCDF /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.0.
↪5.281/netcdf-fortran-4.5.2-wig24pehnu65qiwzipexmjdlbkdupane
$ setenv PNETCDF /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.
↪0.5.281/parallel-netcdf-1.12.1-2djvzsr33qg2cq6lkcsz4bglzjxyzkyv
```

2.3 安装 ecflow

- 下载最新版本的 ecflow, 并按说明安装
- 更新环境变量，测试安装是否成功

```
# add the following two lines to .cshrc or .bashrc

setenv PATH /glade/work/xinzhang/repository/ecflow/bin:$PATH
setenv PYTHONPATH $PYTHONPATH:/glade/work/xinzhang/repository/ecflow/lib/python3.7/
↪site-packages/ecflow
setenv ECF_HOST localhost
setenv ECF_PORT 3141
```

```
$ source ~/.cshrc
$ which ecflow_client
/glade/work/xinzhang/repository/ecflow/bin/ecflow_client
```

- 启动 ecflow

```
ecflow/bin> ecflow_start.sh -p 3141
[03:54:46 14.4.2020] Request( --ping :xinzhang ), Failed to connect to␣
↪cheyenne4:3141. After 2 attempts. Is the server running ?

cheyenne4 cheyenne4 3141
grep: /glade/u/home/xinzhang/.ecflow_ui/servers.txt: No such file or directory
/glade/work/xinzhang/repository/ecflow/bin/ecflow_start.sh: line 165: /glade/u/home/
↪xinzhang/.ecflow_ui/servers.txt: No such file or directory
Tue Apr 14 03:54:46 UTC 2020

User "5591" attempting to start ecf server on "cheyenne4" using ECF_PORT "3141" and␣
↪with:
ECF_HOME     : "/glade/u/home/xinzhang/ecflow_server"
ECF_LOG      : "cheyenne4.3141.ecf.log"
ECF_CHECK    : "cheyenne4.3141.check"
ECF_CHECKOLD : "cheyenne4.3141.check.b"
ECF_OUT      : "/dev/null"

client version is Ecflow version(4.17.2) boost(1.53.0) compiler(gcc 9.2.0)␣
↪protocol(TEXT_ARCHIVE) Compiled on Apr 13 2020 21:33:14
Checking if the server is already running on cheyenne4 and port 3141
[03:54:47 14.4.2020] Request( --ping :xinzhang ), Failed to connect to␣
↪cheyenne4:3141. After 2 attempts. Is the server running ?
```

(下页继续)

(续上页)

```
Backing up check point and log files

OK starting ecFlow server...

Placing server into RESTART mode...

To view server on ecflow_ui - goto Servers/Manage Servers... and enter
Name        : <unique ecFlow server name>
Host        : cheyenne4
Port Number : 3141
```

- 测试 ecflow server 是否启动

```
$ ecflow_client --ping
ping server(localhost:3141) succeeded in 00:00:00.001513  ~1 milliseconds
```

2.4 模块版本信息

在 ${ROOT_DIR}/nwprod/ 下加入各个模块的版本信息。

注解: 当测试新的模块版本的时候，可以在此修改版本号，系统会去相应的目录寻找可执行程序和脚本:

```
nwprod/versions> tree -L 1
.
   hrly_realtime.ver
   hrrr.ver
   obsproc_dump_monitor.ver
   obsproc_rap.ver
   rap.ver

0 directories, 5 files
```

2.5 安装 lib

2.5.1 环境变量设置

在 $HOME/.profile, 或 $HOME/.bashrc, 或 $HOME/.cshrc 中加入以下以下与库相关的环境变量:

> **注解:** jasper version 2.0.0 above is not working, please install version 1.900.1

```
# Define the nwp libs
# -------------------------

setenv BACIO_VER v2.0.1
setenv SIGIO_VER v2.0.1
setenv W3EMC_VER v2.2.0
setenv SFCIO_VER v1.0.0

# Defin the include paths
# -------------------------

setenv JASPER_INC /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.
↪0.5.281/jasper-1.900.1-q4jmxkhrtqgr5linm2ekuaiglaydhoej/include
setenv PNG_INC /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.0.
↪5.281/libpng-1.6.37-rty7tjtcgxiwgolsdko54yrouft5aaqd/include
setenv Z_INC /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.0.5.
↪281/zlib-1.2.11-ezdt444oqtuowpfngoi3ibcwjzgzatfd/include

setenv XMLPARSE_INC ${ROOT_DIR}/nwprod/lib/xmlparse/v2.0.0/intel/xmlparse-2.0.0/include
setenv G2TMPL_INC ${ROOT_DIR}/nwprod/lib/g2tmpl/v1.4.0/intel/include
setenv G2_INC4 ${ROOT_DIR}/nwprod/lib/g2/v2.5.2/intel/include/g2_4
setenv G2_INCd ${ROOT_DIR}/nwprod/lib/g2/v2.5.2/intel/include/g2_d
setenv NEMSIO_INC ${ROOT_DIR}/nwprod/lib/nemsio/incmod/nemsio
setenv GFSIO_INC4 ${ROOT_DIR}/nwprod/lib/gfsio/v1.1.0/intel/include/gfsio_4
setenv SIGIO_INC4 ${ROOT_DIR}/nwprod/lib/sigio/v2.0.1/src/sigio_v2.0.1/include
setenv SFCIO_INC4 ${ROOT_DIR}/nwprod/lib/sfcio/v1.0.0/intel/include/sfcio_v1.0.0_4
setenv W3EMC_INC4 ${ROOT_DIR}/nwprod/lib/w3emc/v2.2.0/src/intel/w3emc_v2.2.0_4
setenv W3EMC_INCd ${ROOT_DIR}/nwprod/lib/w3emc/v2.2.0/src/intel/w3emc_v2.2.0_d
setenv CRTM_INC ${ROOT_DIR}/nwprod/lib/crtm/v2.2.3/intel/include/crtm_v2.2.3/

# Defin the library paths
```

(下页继续)

(续上页)

```
# -------------------------

setenv JASPER_LIB "-L/glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-
↪19.0.5.281/jasper-1.900.1-q4jmxkhrtqgr5linm2ekuaiglaydhoej/lib -ljasper"
setenv PNG_LIB "-L/glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.
↪0.5.281/libpng-1.6.37-rty7tjtcgxiwgolsdko54yrouft5aaqd/lib -lpng"
setenv Z_LIB "-L/glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.0.
↪5.281/zlib-1.2.11-ezdt444oqtuowpfngoi3ibcwjzgzatfd/lib -lz"

setenv XMLPARSE_LIB ${ROOT_DIR}/nwprod/lib/xmlparse/v2.0.0/intel/xmlparse-2.0.0/lib/
↪libxmlparse.a
setenv G2TMPL_LIB ${ROOT_DIR}/nwprod/lib/g2tmpl/v1.4.0/intel/libg2tmpl.a
setenv G2_LIB4 ${ROOT_DIR}/nwprod/lib/g2/v2.5.2/intel/libg2_4.a
setenv G2_LIBd ${ROOT_DIR}/nwprod/lib/g2/v2.5.2/intel/libg2_d.a
setenv NEMSIO_LIB ${ROOT_DIR}/nwprod/lib/nemsio/libnemsio.a
setenv GFSIO_LIB4 ${ROOT_DIR}/nwprod/lib/gfsio/v1.1.0/intel/libgfsio_4.a
setenv SIGIO_LIB4 ${ROOT_DIR}/nwprod/lib/sigio/v2.0.1/src/sigio_v2.0.1/lib/libsigio_v2.0.
↪1_4.a
setenv SFCIO_LIB4 ${ROOT_DIR}/nwprod/lib/sfcio/v1.0.0/intel/libsfcio_v1.0.0_4.a
setenv IP_LIB ${ROOT_DIR}/nwprod/lib/ip/v2.0.0/intel/libip_v2.0.0_4.a
setenv IP_LIB4 ${ROOT_DIR}/nwprod/lib/ip/v2.0.0/intel/libip_v2.0.0_4.a
setenv SP_LIB4 ${ROOT_DIR}/nwprod/lib/sp/libsp_4.a
setenv SP_LIBd ${ROOT_DIR}/nwprod/lib/sp/libsp_d.a
setenv W3NCO_LIB4 ${ROOT_DIR}/nwprod/lib/w3nco/v2.0.6/intel/libw3nco_v2.0.1_4.a
setenv W3NCO_LIB8 ${ROOT_DIR}/nwprod/lib/w3nco/v2.0.6/intel/libw3nco_v2.0.1_8.a
setenv W3NCO_LIBd ${ROOT_DIR}/nwprod/lib/w3nco/v2.0.6/intel/libw3nco_v2.0.1_d.a
setenv W3EMC_LIB4 ${ROOT_DIR}/nwprod/lib/w3emc/v2.2.0/src/intel/libw3emc_v2.2.0_4.a
setenv W3EMC_LIB8 ${ROOT_DIR}/nwprod/lib/w3emc/v2.2.0/src/intel/libw3emc_v2.2.0_8.a
setenv W3EMC_LIBd ${ROOT_DIR}/nwprod/lib/w3emc/v2.2.0/src/intel/libw3emc_v2.2.0_d.a
setenv BACIO_LIB4 ${ROOT_DIR}/nwprod/lib/bacio/v2.0.1/intel/libbacio_4.a
setenv BACIO_LIB8 ${ROOT_DIR}/nwprod/lib/bacio/v2.0.1/intel/libbacio_8.a
setenv CRTM_LIB ${ROOT_DIR}/nwprod/lib/crtm/v2.2.3/intel/libcrtm_v2.2.3.a
setenv BUFR_LIB4_DA ${ROOT_DIR}/nwprod/lib/bufr/v11.0.1/intel/libbufr_v11.0.1_4_64_DA.a
setenv BUFR_LIB4 ${BUFR_LIB4_DA}
setenv BUFR_LIB8_DA ${ROOT_DIR}/nwprod/lib/bufr/v11.0.1/intel/libbufr_v11.0.1_8_64_DA.a
setenv BUFR_LIB8 ${BUFR_LIB8_DA}
setenv BUFR_LIBd_DA ${ROOT_DIR}/nwprod/lib/bufr/v11.0.1/intel/libbufr_v11.0.1_d_64_DA.a
setenv BUFR_LIBd ${BUFR_LIBd_DA}
```

2.5.2 bacio

```
setenv COMP intel
cd ${ROOT_DIR}/nwprod/lib/bacio/v2.0.1/src
chmod +x makebacio_wcoss.sh
rm -f ../intel/libbacio_*.a
./makebacio_wcoss.sh

v2.0.1/src> tree -L 1 ../intel/
../intel/
   libbacio_4.a
   libbacio_8.a

0 directories, 2 files
```

2.5.3 bufr

```
setenv COMP intel
cd ${ROOT_DIR}/nwprod/lib/bufr/v11.0.1/src
rm -fr ../intel/libbufr*.a
chmod +x makebufrlib.sh
./clean.sh
./makebufrlib.sh

v11.0.1/src> tree -L 1 ../intel/
../intel/
   libbufr_v11.0.1_4_64_DA.a
   libbufr_v11.0.1_8_64_DA.a
   libbufr_v11.0.1_d_64_DA.a

0 directories, 3 files
```

2.5.4 g2

```
setenv COMP intel
setenv JASPER_INC /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-
↪broadwell/intel-19.0.5.281/jasper-1.900.1-q4jmxkhrtqgr5linm2ekuaiglaydhoej/
↪include
setenv PNG_INC /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/
↪intel-19.0.5.281/libpng-1.6.37-rty7tjtcgxiwgolsdko54yrouft5aaqd/include
```

(下页继续)

(续上页)

```
setenv Z_INC /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/
↪intel-19.0.5.281/zlib-1.2.11-ezdt444oqtuowpfngoi3ibcwjzgzatfd/include

cd ${ROOT_DIR}/nwprod/lib/g2/v2.5.2/src
chmod +x makeg2lib.sh
rm ../intel/*.a
./makeg2lib.sh

v2.5.2/src> tree -L 3 ../intel/
../intel/
   include
       g2_4
           drstemplates.mod
           g2grids.mod
           grib_mod.mod
           gridtemplates.mod
           params_ecmwf.mod
           params.mod
           pdstemplates.mod
           re_alloc.mod
       g2_d
            drstemplates.mod
            g2grids.mod
            grib_mod.mod
            gridtemplates.mod
            params_ecmwf.mod
            params.mod
            pdstemplates.mod
            re_alloc.mod
   libg2_4.a
   libg2_d.a

3 directories, 18 files
```

2.5.5 g2tmpl

```
setenv COMP intel
cd ${ROOT_DIR}/nwprod/lib/g2tmpl/v1.4.0/src
chmod +x makeg2tmpllib.sh
```

(下页继续)

(续上页)

```
rm -f ../intel/libg2tmpl.a
./makeg2tmpllib.sh

v1.4.0/src> tree -L 2 ../intel/
../intel/
   include
       gdtsec3.mod
       grib2_all_tables_module.mod
   libg2tmpl.a

1 directory, 3 files
```

2.5.6 gfsio

```
setenv COMP intel
cd ${ROOT_DIR}/nwprod/lib/gfsio/v1.1.0/src
chmod +x make_wcoss.sh
rm -f ../intel/libgfsio_4.a
./make_wcoss.sh

v1.1.0/src> tree -L 3 ../intel/
../intel/
   include
       gfsio_4
           gfsio_module.mod
   libgfsio_4.a

2 directories, 2 files
```

2.5.7 ip

```
setenv COMP intel
setenv IP_VER v2.0.0
cd ${ROOT_DIR}/nwprod/lib/ip/v2.0.0/src
cp makefile.wcoss.conf makefile.conf
chmod +x makelibip.sh
./makelibip.sh
```

(下页继续)

(续上页)

```
v2.0.0/src> tree -L 1  ../intel/
../intel/
   libip_v2.0.0_4.a
   libip_v2.0.0_8.a
   libip_v2.0.0_d.a

0 directories, 3 files



cd ${ROOT_DIR}/nwprod/lib/ip/v3.0.0
./make_ip_lib_wcoss-cray.sh ifort.setup

ip/v3.0.0> tree -L 3 ip/v3.0.0/intel/
ip/v3.0.0/intel/
   config.log.ip_v3.0.0_4
   config.log.ip_v3.0.0_8
   config.log.ip_v3.0.0_d
   include
       ip_v3.0.0_4
           gdswzd00_mod.mod
           gdswzd01_mod.mod
           gdswzd03_mod.mod
           gdswzd04_mod.mod
           gdswzd05_mod.mod
           gdswzdcb_mod.mod
           gdswzdcd_mod.mod
           gdswzd_mod.mod
           iplib.h
       ip_v3.0.0_8
           gdswzd00_mod.mod
           gdswzd01_mod.mod
           gdswzd03_mod.mod
           gdswzd04_mod.mod
           gdswzd05_mod.mod
           gdswzdcb_mod.mod
           gdswzdcd_mod.mod
           gdswzd_mod.mod
           iplib.h
       ip_v3.0.0_d
            gdswzd00_mod.mod
```

(下页继续)

(续上页)

```
          gdswzd01_mod.mod
          gdswzd03_mod.mod
          gdswzd04_mod.mod
          gdswzd05_mod.mod
          gdswzdcb_mod.mod
          gdswzdcd_mod.mod
          gdswzd_mod.mod
          iplib.h
   libip_v3.0.0_4.a
   libip_v3.0.0_8.a
   libip_v3.0.0_d.a

4 directories, 33 files
```

2.5.8 nemsio

```
cd ${ROOT_DIR}/nwprod/lib/nemsio/v2.2.2/src
cp conf/configure.Linux.intel conf/configure
chmod +x compile
rm -f *.mod *.o
./compile

v2.2.2/src> tree -L 3 ../..
../..
   incmod
       nemsio
           nemsio_module.mod
           nemsio_module_mpi.mod
           nemsio_openclose.mod
           nemsio_read.mod
           nemsio_write.mod
   libnemsio.a
   v2.2.2
       src
           compile
           compile-cray.log
           compile-intel.log
           conf
           FOR_SPA
```

(下页继续)

(续上页)

```
           incmod
           Makefile
           nemsio_module.f90
           nemsio_module_mpi.f90
           nemsio_openclose.f90
           nemsio_read.f90
           nemsio_write.f90
           readme

6 directories, 17 files
```

2.5.9 sfcio

```
setenv COMP intel
setenv SFCIO_VER v1.0.0
cd ${ROOT_DIR}/nwprod/lib/sfcio/v1.0.0/src
rm ../../libsfcio_big_4.a
chmod +x make.sh
./make.sh

v1.0.0/src> tree -L 3 ../intel/
../intel/
   include
       sfcio_v1.0.0_4
           sfcio_module.mod
   libsfcio_v1.0.0_4.a

2 directories, 2 files
```

2.5.10 sigio

```
setenv COMP intel
setenv SIGIO_VER v2.0.1
cd ${ROOT_DIR}/nwprod/lib/sigio/v2.0.1/src
make clean
make
touch ../config.log
make install
```

(下页继续)

(续上页)

```
v2.0.1/src> tree -L 2 sigio_v2.0.1/
sigio_v2.0.1/
   config.log
   include
       sigio_module.mod
       sigio_r_module.mod
   lib
        libsigio_v2.0.1_4.a

2 directories, 4 files
```

2.5.11 sp

```
cd ${ROOT_DIR}/nwprod/lib/sp/v2.0.2/src
chmod +x makelibsp.sh_Linux
rm ../../libsp_*a
./makelibsp.sh_Linux

v2.0.2/src> tree -L 1 ../..
../..
   libsp_4.a
   libsp_8.a
   libsp_d.a
   v2.0.2
```

2.5.12 w3emc

```
setenv COMP intel
setenv W3EMC_VER v2.2.0
cd ${ROOT_DIR}/nwprod/lib/w3emc/v2.2.0/src
setenv W3EMC_SRC `pwd`
chmod +x configure
chmod +x make_w3emc_lib.sh
./make_w3emc_lib.sh ifort.setup

v2.2.0/src> tree -L 2 intel/
intel/
```

(下页继续)

(续上页)

```
  libw3emc_v2.2.0_4.a
  libw3emc_v2.2.0_8.a
  libw3emc_v2.2.0_d.a
  w3emc_v2.2.0_4
      args_mod.mod
      gblevn_module.mod
      mersenne_twister.mod
  w3emc_v2.2.0_8
      args_mod.mod
      gblevn_module.mod
      mersenne_twister.mod
  w3emc_v2.2.0_d
       args_mod.mod
       gblevn_module.mod
       mersenne_twister.mod

3 directories, 12 files
```

2.5.13 w3nco

```
setenv COMP intel
cd ${ROOT_DIR}/nwprod/lib/w3nco/v2.0.6/src
rm ../intel/libw3nco_*.a
chmod +x makelibw3_nco.sh
./makelibw3_nco.sh

v2.0.6/src> tree -L 1 ../intel/
../intel/
  libw3nco_v2.0.1_4.a
  libw3nco_v2.0.1_8.a
  libw3nco_v2.0.1_d.a

0 directories, 3 files
```

2.5.14 xmlparse

```
cd ${ROOT_DIR}/nwprod/lib/xmlparse/v2.0.0/src
setenv COMP intel
```

(下页继续)

(续上页)

```
make clean
./configure --prefix=${ROOT_DIR}/nwprod/lib/xmlparse/v2.0.0/${COMP} FC=ifort
make clean
make
make install

v2.0.0/src> tree -L 3 ../intel/
../intel/
   xmlparse-2.0.0
       include
           read_xml_primitives.mod
           xmlparse.mod
       lib
            libxmlparse.a

3 directories, 3 files
```

2.5.15 crtm

```
# Fix files are used by wrfpost run

cd ${ROOT_DIR}/nwprod/lib/crtm/v2.0.6/src
chmod +x configure/ifort.csh.setup
source configure/ifort.csh.setup
make clean
make
mkdir -p ../intel/include/crtm_v2.0.6
mv libsrc/libCRTM.a ../intel/libcrtm_v2.0.6.a
mv libsrc/*.mod ../intel/include/crtm_v2.0.6/

v2.0.6/src> tree -L 3 ../intel/
../intel/
   include
       crtm_v2.0.6
           aerosolcoeff_binary_io.mod
           aerosolcoeff_define.mod
           antcorr_binary_io.mod
           antcorr_define.mod
           asvar_define.mod
```

(下页继续)

(续上页)

```
becoeff_binary_io.mod
becoeff_define.mod
binary_file_utility.mod
cloudcoeff_binary_io.mod
cloudcoeff_define.mod
compare_float_numbers.mod
crtm_adjoint_module.mod
crtm_aerosolcoeff.mod
crtm_aerosol_define.mod
crtm_aerosol_io.mod
crtm_aerosolscatter.mod
crtm_ancillaryinput_define.mod
crtm_antcorr.mod
crtm_atmabsorption.mod
crtm_atmoptics.mod
crtm_atmosphere_define.mod
crtm_atmosphere_io.mod
crtm_atmosphere.mod
crtm_atmscatter_define.mod
crtm_becoeff.mod
crtm_channelinfo_define.mod
crtm_cloudcoeff.mod
crtm_cloud_define.mod
crtm_cloud_io.mod
crtm_cloudscatter.mod
crtm_emiscoeff.mod
crtm_fastem1.mod
crtm_fastem3.mod
crtm_forward_module.mod
crtm_geometry_define.mod
crtm_geometryinfo_define.mod
crtm_geometryinfo.mod
crtm_geometry_io.mod
crtm_interpolation.mod
crtm_ir_ice_sfcoptics.mod
crtm_ir_land_sfcoptics.mod
crtm_ir_snow_sfcoptics.mod
crtm_irssem.mod
crtm_ir_water_sfcoptics.mod
crtm_k_matrix_module.mod
```

(下页继续)

(续上页)

```
        crtm_lifecycle.mod
        crtm_lowfrequency_mwssem.mod
        crtm_model_profiles.mod
        crtm_module.mod
        crtm_moleculescatter.mod
        crtm_mw_ice_sfcoptics.mod
        crtm_mw_land_sfcoptics.mod
        crtm_mw_snow_sfcoptics.mod
        crtm_mw_water_sfcoptics.mod
        crtm_options_define.mod
        crtm_parameters.mod
        crtm_planck_functions.mod
        crtm_rtsolution_define.mod
        crtm_rtsolution_io.mod
        crtm_rtsolution.mod
        crtm_sensordata_define.mod
        crtm_sensordata_io.mod
        crtm_sensorinfo.mod
        crtm_sfcoptics_define.mod
        crtm_sfcoptics.mod
        crtm_spccoeff.mod
        crtm_surface_define.mod
        crtm_surface_io.mod
        crtm_surface_ir_emissivity.mod
        crtm_tangent_linear_module.mod
        crtm_taucoeff.mod
        crtm_utility.mod
        crtm_vis_water_sfcoptics.mod
        csvar_define.mod
        date_utility.mod
        emiscoeff_binary_io.mod
        emiscoeff_define.mod
        endian_utility.mod
        file_utility.mod
        fresnel.mod
        fundamental_constants.mod
        iatm_define.mod
        message_handler.mod
        nesdis_amsre_siceem_module.mod
        nesdis_amsre_snowem_module.mod
```

(下页继续)

(续上页)

```
        nesdis_amsu_siceem_module.mod
        nesdis_amsu_snowem_module.mod
        nesdis_landem_module.mod
        nesdis_mhs_siceem_module.mod
        nesdis_mhs_snowem_module.mod
        nesdis_oceanem_module.mod
        nesdis_seaice_phyem_module.mod
        nesdis_snowem_parameters.mod
        nesdis_ssmi_module.mod
        nesdis_ssmi_siceem_module.mod
        nesdis_ssmi_snowem_module.mod
        nesdis_ssmis_seaiceem_module.mod
        nesdis_ssmis_snowem_module.mod
        ocean_permittivity.mod
        odas_atmabsorption.mod
        odas_binary_io.mod
        odas_define.mod
        odas_predictor_define.mod
        odas_predictor.mod
        odas_taucoeff.mod
        odps_atmabsorption.mod
        odps_binary_io.mod
        odps_coordinatemapping.mod
        odps_define.mod
        odps_predictor_define.mod
        odps_predictor.mod
        odps_taucoeff.mod
        odssu_atmabsorption.mod
        odssu_binary_io.mod
        odssu_define.mod
        odssu_taucoeff.mod
        odzeeman_atmabsorption.mod
        odzeeman_predictor.mod
        odzeeman_taucoeff.mod
        profile_utility_parameters.mod
        rtv_define.mod
        search_utility.mod
        sensorinfo_parameters.mod
        sort_utility.mod
        spccoeff_binary_io.mod
```

(下页继续)

(续上页)

```
          spccoeff_define.mod
          spectral_units_conversion.mod
          ssu_input_define.mod
          taucoeff_define.mod
          type_kinds.mod
          zeeman_input_define.mod
          zeeman_utility.mod
   libcrtm_v2.0.6.a

2 directories, 133 files


# Used by compiling gsi and wrfpost

cd ${ROOT_DIR}/nwprod/lib/crtm/v2.2.3/src
chmod +x make_crtm_lib.sh
chmod +x configure
rm -fr ../intel
./make_crtm_lib.sh ifort.setup
mkdir -p ../intel/include/crtm_v2.2.3
mv libsrc/libcrtm.a ../intel/libcrtm_v2.2.3.a
mv libsrc/*.mod ../intel/include/crtm_v2.2.3/.
chmod +x link_crtm_coeffs.sh
./link_crtm_coeffs.sh fix ${ROOT_DIR}/nwprod/lib/crtm/v2.2.3/fix

v2.2.3/src> tree -L 3 ../intel/
../intel/
   include
       crtm_v2.2.3
           accoeff_binary_io.mod
           accoeff_define.mod
           ada_module.mod
           aerosolcoeff_binary_io.mod
           aerosolcoeff_define.mod
           aovar_define.mod
           asvar_define.mod
           azimuth_emissivity_f6_module.mod
           azimuth_emissivity_module.mod
           binary_file_utility.mod
           cloudcoeff_binary_io.mod
```

(下页继续)

(续上页)

```
        cloudcoeff_define.mod
        common_rtsolution.mod
        compare_float_numbers.mod
        crtm_adjoint_module.mod
        crtm_aerosolcoeff.mod
        crtm_aerosol_define.mod
        crtm_aerosolscatter.mod
        crtm_ancillaryinput_define.mod
        crtm_antennacorrection.mod
        crtm_aod_module.mod
        crtm_atmabsorption.mod
        crtm_atmoptics_define.mod
        crtm_atmoptics.mod
        crtm_atmosphere_define.mod
        crtm_atmosphere.mod
        crtm_channelinfo_define.mod
        crtm_cloudcoeff.mod
        crtm_cloud_define.mod
        crtm_cloudscatter.mod
        crtm_fastem1.mod
        crtm_fastemx.mod
        crtm_forward_module.mod
        crtm_geometry_define.mod
        crtm_geometryinfo_define.mod
        crtm_geometryinfo.mod
        crtm_interpolation.mod
        crtm_iricecoeff.mod
        crtm_ir_ice_sfcoptics.mod
        crtm_irlandcoeff.mod
        crtm_ir_land_sfcoptics.mod
        crtm_irsnowcoeff.mod
        crtm_ir_snow_sfcoptics.mod
        crtm_irssem.mod
        crtm_irwatercoeff.mod
        crtm_ir_water_sfcoptics.mod
        crtm_k_matrix_module.mod
        crtm_lifecycle.mod
        crtm_lowfrequency_mwssem.mod
        crtm_model_profiles.mod
        crtm_module.mod
```

(下页继续)

(续上页)

```
        crtm_moleculescatter.mod
        crtm_mw_ice_sfcoptics.mod
        crtm_mw_land_sfcoptics.mod
        crtm_mw_snow_sfcoptics.mod
        crtm_mwwatercoeff.mod
        crtm_mw_water_sfcoptics.mod
        crtm_nltecorrection.mod
        crtm_options_define.mod
        crtm_parameters.mod
        crtm_planck_functions.mod
        crtm_predictor_define.mod
        crtm_predictor.mod
        crtm_rtsolution_define.mod
        crtm_rtsolution.mod
        crtm_secategory.mod
        crtm_sensordata_define.mod
        crtm_sensorinfo.mod
        crtm_sfcoptics_define.mod
        crtm_sfcoptics.mod
        crtm_spccoeff.mod
        crtm_surface_define.mod
        crtm_tangent_linear_module.mod
        crtm_taucoeff.mod
        crtm_utility.mod
        crtm_visicecoeff.mod
        crtm_vis_ice_sfcoptics.mod
        crtm_vislandcoeff.mod
        crtm_vis_land_sfcoptics.mod
        crtm_vissnowcoeff.mod
        crtm_vis_snow_sfcoptics.mod
        crtm_viswatercoeff.mod
        crtm_vis_water_sfcoptics.mod
        csvar_define.mod
        datetime_utility.mod
        date_utility.mod
        ellison.mod
        emission_module.mod
        endian_utility.mod
        file_utility.mod
        fitcoeff_define.mod
```

(下页继续)

(续上页)

```
        foam_utility_module.mod
        fresnel.mod
        fundamental_constants.mod
        guillou.mod
        hyperbolic_step.mod
        iatm_define.mod
        irwatercoeff_define.mod
        large_scale_correction_module.mod
        liu.mod
        lseatlas_define.mod
        message_handler.mod
        mwwatercoeff_define.mod
        mwwaterlut_define.mod
        nesdis_amsre_siceem_module.mod
        nesdis_amsre_snowem_module.mod
        nesdis_amsu_siceem_module.mod
        nesdis_amsu_snowem_module.mod
        nesdis_atms_snowem_module.mod
        nesdis_landem_module.mod
        nesdis_mhs_siceem_module.mod
        nesdis_mhs_snowem_module.mod
        nesdis_seaice_phyem_module.mod
        nesdis_snowem_atms_parameters.mod
        nesdis_snowem_parameters.mod
        nesdis_ssmi_module.mod
        nesdis_ssmi_siceem_module.mod
        nesdis_ssmi_snowem_module.mod
        nesdis_ssmis_seaiceem_module.mod
        nesdis_ssmis_snowem_module.mod
        nltecoeff_binary_io.mod
        nltecoeff_define.mod
        nlte_parameters.mod
        nlte_predictor_define.mod
        nlte_predictor_io.mod
        odas_atmabsorption.mod
        odas_binary_io.mod
        odas_define.mod
        odas_predictor_define.mod
        odas_predictor.mod
        odas_taucoeff.mod
```

(下页继续)

(续上页)

```
        odps_atmabsorption.mod
        odps_binary_io.mod
        odps_coordinatemapping.mod
        odps_define.mod
        odps_predictor_define.mod
        odps_predictor.mod
        odps_taucoeff.mod
        odssu_atmabsorption.mod
        odssu_binary_io.mod
        odssu_define.mod
        odssu_taucoeff.mod
        odzeeman_atmabsorption.mod
        odzeeman_predictor.mod
        odzeeman_taucoeff.mod
        pafv_define.mod
        profile_utility_parameters.mod
        reflection_correction_module.mod
        rtv_define.mod
        search_utility.mod
        secategory_define.mod
        sensorinfo_parameters.mod
        slope_variance.mod
        small_scale_correction_module.mod
        soi_module.mod
        sort_utility.mod
        spccoeff_binary_io.mod
        spccoeff_define.mod
        spectral_units_conversion.mod
        ssu_input_define.mod
        string_utility.mod
        subset_define.mod
        taucoeff_define.mod
        timing_utility.mod
        type_kinds.mod
        unittest_define.mod
        zeeman_input_define.mod
        zeeman_utility.mod
  libcrtm_v2.2.3.a

2 directories, 169 files
```

注解:

1. Fix files are used by gsi run
2. Do not need to build at this moment

```
cd ${ROOT_DIR}/nwprod/lib/crtm/v2.2.6/src
./link_crtm_coeffs.sh fix ${ROOT_DIR}/nwprod/lib/crtm/v2.2.6/fix
```

2.6 安装 gempak

- 下载最新版本的 gempark

```
mkdir -p ${ROOT_DIR}/nwprod/gempak
cd ${ROOT_DIR}/nwprod/gempak
tar xvf gempak-7.5.1.tar ; rm gempak-7.5.1.tar
ln -s GEMPAK7/ NAWIPS
cd ${ROOT_DIR}/nwprod/gempak/NAWIPS
```

- edit `Gemenviron`

```
setenv NAWIPS ${ROOT_DIR}/nwprod/gempak/GEMPAK7
```

注解: libxslt need to be installed Python version 3 is not working likely

- compile

```
source Gemenviron
make everything |& tee make.log
```

2.7 安装 decoders

注解: need to add following lines in `Gemenviron.profile` or `Gemenviron`

```
GEMINC=$GEMPAK/include
export GEMINC
```

```
# Use gempak envirs
# ------------------------

cd ${ROOT_DIR}/nwprod/gempak
cp NAWIPS/Gemenviron .gempak
chmod +x .gempak
source .gempak
cd ${ROOT_DIR}/nwprod/decoders/decod_shared/lib

# Compile crex
# -------------------------

cd crex
rm libcrex.a
./makecrex.sh
cd ..
ln -sf crex/libcrex.a libcrex.a

# Compile madis
# ---------------------------

cd madis/src
make clean
make
cd ../..
ln -sf madis/lib/madislib.a libmadis.a

# Compile decod_ut
# ---------------------------

cd decod_ut
rm libdecod_ut.a
./makedecod_ut.sh
cd ..
ln -sf decod_ut/libdecod_ut.a libdecod_ut.a

# Compile bridge
# ---------------------------

cd bridge
```

(下页继续)

(续上页)

```
rm libbridge.a
./makebridge.sh
cd ..
ln -fs bridge/libbridge.a libbridge.a

ln -fs ${ROOT_DIR}/nwprod/gempak/NAWIPS/os/linux64/lib/rsl.a librsl.a

# Compile decoders
# ---------------------------

cd ${ROOT_DIR}/nwprod/gempak/NAWIPS/os/linux64/lib
ln -fs gemlib.a libgemlib.a
ln -fs cgemlib.a libcgemlib.a
ln -fs gplt.a libgplt.a
ln -fs gn.a libgn.a
ln -fs device.a libdevice.a

cd ${ROOT_DIR}/nwprod/decoders/decod_shared/include
ln -fs ${ROOT_DIR}/nwprod/gempak/NAWIPS/Gemenviron.profile .gempak
chmod +x make_decoder_links.sh
./make_decoder_links.sh
chmod +x make_all_decoders.sh
./make_all_decoders.sh
ecod_dccrad is not compiled due to the rsl.a
```

2.8 安装 obsproc

2.8.1 环境变量的设置

```
setenv obsproc_dump_ver v5.0.1
setenv obsproc_dump_post_ver v3.3.1
setenv obsproc_prep_ver v5.2.1
setenv obsproc_prep_post_ver v3.1.0
setenv obsproc_rap_ver v3.1.0
setenv obsproc_shared_bufr_remorest_ver v2.1.0
setenv obsproc_shared_bufr_dumplist_ver v2.0.2
```

2.8.2 obsproc_shared

```
cd ${ROOT_DIR}/nwprod/obsproc_shared/bufr_avgdata.v2.1.0/sorc
./clean.sh
./build.sh
./install.sh

cd ${ROOT_DIR}/nwprod/obsproc_shared/bufr_cword.v1.0.0/sorc
./build.sh
./install.sh

cd ${ROOT_DIR}/nwprod/obsproc_shared/bufr_remorest.${obsproc_shared_bufr_remorest_ver}/
↪sorc
./clean.sh
./build.sh
./install.sh
```

2.8.3 obsproc_dump

```
cd ${ROOT_DIR}/nwprod/obsproc_dump.${obsproc_dump_ver}/sorc
chmod +x clobber.sh
./clobber.sh
chmod +x build.sh
./build.sh
chmod +x install.sh
./install.sh

obsproc_dump.v5.0.1/sorc> tree -L 1 ../exec
../exec
   bufr_chkbfr
   bufr_combfr
   bufr_dcodwindsat
   bufr_dumpmd
   bufr_dupair
   bufr_dupcor
   bufr_dupmar
   bufr_dupmrg
   bufr_duprad
   bufr_dupsat
   bufr_dupshp
```

(下页继续)

(续上页)

```
   bufr_dupsst
   bufr_edtbfr
   bufr_geofil
   bufr_quipc
   bufr_raddate
   bufr_supertmi
   prepobs_prepssmi
   wave_dcodquikscat

0 directories, 19 files
```

2.8.4 obsproc_dump_post

```
cd ${ROOT_DIR}/nwprod/obsproc_dump_post.${obsproc_dump_post_ver}/sorc
mkdir -p ../exec
rm -fr ../exec/*
chmod +x clean.sh
./clean.sh
cd bufr_datacount.fd
make
make install
cd -

cd bufr_listdumps.fd
make
make install

cd ..
obsproc_dump_post.v3.3.1/sorc> tree -L 1 ../exec/
../exec/
   bufr_datacount
   bufr_listdumps

0 directories, 2 files
```

2.8.5 obsproc_prep

```
cd ${ROOT_DIR}/nwprod/obsproc_prep.${obsproc_prep_ver}/sorc
mkdir -p ../exec
rm ../exec/*
chmod +x clean.sh
./clean.sh
chmod +x build.sh
./build.sh
chmod +x install.sh
./install.sh

obsproc_prep.v5.2.1/sorc> tree -L 1 ../exec/
../exec/
   prepobs_cqcbufr
   prepobs_cqcvad
   prepobs_glerladj
   prepobs_listheaders
   prepobs_monoprepbufr
   prepobs_mpcopybufr
   prepobs_oiqcbufr
   prepobs_prepacpf
   prepobs_prepacqc
   prepobs_prepanow
   prepobs_prepdata
   prepobs_prevents
   prepobs_profcqc
   syndat_syndata

0 directories, 14 files
```

2.8.6 obsproc_prep_post

```
cd ${ROOT_DIR}/nwprod/obsproc_prep_post.${obsproc_prep_post_ver}/sorc
mkdir -p ../exec
rm -f ../exec/*
chmod +x clean.sh
./clean.sh

cd gdascounts_ave.fd
```

(下页继续)

(续上页)

```
make
make install
cd -

cd global_postevents.fd
make
make install
cd -

cd timetwin.fd
make
make install
cd -

obsproc_prep_post.v3.1.0/sorc> tree -L 1 ../exec/
../exec/
   gdascounts_ave
   global_postevents
   timetwin

0 directories, 3 files
```

2.9 安装 prod_util

2.9.1 安装步骤

编译 fsync_file.cd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/prod_util.v1.1.3/sorc/fsync_file.cd
rm -f fsync_file
make
mv fsync_file ${ROOT_DIR}/nwprod/prod_util.v1.1.3/ush/.
```

编译 mdate.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/prod_util.v1.1.3/sorc/mdate.fd
rm -f mdate
make
mv mdate ${ROOT_DIR}/nwprod/prod_util.v1.1.3/ush/.
```

编译 ndate

编译步骤:

```
cd ${ROOT_DIR}/nwprod/prod_util.v1.1.3/sorc/ndate.fd
rm -f ndate
make
mv ndate ${ROOT_DIR}/nwprod/prod_util.v1.1.3/ush/.
```

编译 nhour

编译步骤:

```
cd ${ROOT_DIR}/nwprod/prod_util.v1.1.3/sorc/nhour.fd
rm -f nhour
make
mv nhour ${ROOT_DIR}/nwprod/prod_util.v1.1.3/ush/.
```

更新 ~/.profile

```
export PATH=${ROOT_DIR}/nwprod/prod_util.v1.1.3/ush:$PATH
```

2.10 安装 grib_util

2.10.1 安装步骤

编译 copygb2.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/copygb2.fd
setenv IP_INCd ${ROOT_DIR}/nwprod/lib/ip/v3.0.0/ip/v3.0.0/intel/include/ip_v3.
↪0.0_d
```

(下页继续)

(续上页)

```
setenv IP_LIBd ${ROOT_DIR}/nwprod/lib/ip/v3.0.0/ip/v3.0.0/intel/libip_v3.0.0_d.
↪a
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 degrib2.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/degrib2.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 grbindex.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/grbindex.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 tocgrib2.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/tocgrib2.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 tocgrib.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/tocgrib.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 cnvgrib.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/cnvgrib.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 copygb.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/copygb.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 grb2index.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/grb2index.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 grib2grib.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/grib2grib.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 tocgrib2super.fd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/tocgrib2super.fd
cp makefile_wcoss_theia makefile
make clean
make
make install
```

编译 wgrib.cd

编译步骤:

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/sorc/wgrib.cd
cp makefile_wcoss_theia makefile
rm wgrib
make
make install
```

编译 wgrib2

必须使用较新版本的 wgrib2 , 下载 wgrib2.v2.0.8

```
cd ${ROOT_DIR}/nwprod/grib_util.v1.1.1/
mv wgrib2.tgz.v2.0.8 wgrib2.tgz
tar xvf wgrib2.tgz
mv grib2 wgrib2.cd
cd wgrib2.cd
setenv CC gcc
```

(下页继续)

(续上页)

```
setenv CPP /usr/bin/cpp
setenv FC gfortran
make
cp wgrib2/wgrib2 ../../exec/.
```

```
grib_util.v1.1.1/sorc> tree -L 1 ../exec/
../exec/
   cnvgrib
   copygb
   copygb2
   degrib2
   grb2index
   grbindex
   grib2grib
   tocgrib
   tocgrib2
   tocgrib2super
   wgrib
   wgrib2

0 directories, 12 files
```

2.11 安装 bufr_util

2.11.1 安装步骤

编译 bufr_tranjb

编译步骤:

```
cd ${ROOT_DIR}/nwprod/bufr_util/sorc/bufr_tranjb.fd
mkdir -p ${ROOT_DIR}/nwprod/bufr_util/exc
make clean
make
make install
```

编译 cwordsh

编译步骤:

```
cd ${ROOT_DIR}/nwprod/bufr_util/sorc/cwordsh.fd
make clean
make
mv cwordsh ../../exec/.
```

编译 debufr

编译步骤:

```
cd ${ROOT_DIR}/nwprod/bufr_util/sorc/debufr.fd
make clean
make
make install
```

```
cd ${ROOT_DIR}/nwprod/bufr_util/exec
bufr_util/exec> tree -L 1
.
   bufr_tranjb
   cwordsh
   debufr

0 directories, 3 files
```

2.12 安装 rap

2.12.1 公用环境变量

将如下环境变量加入到 ~/.bashrc 或 ~/.cshrc 或 ~/.profile 中

```
setenv NETCDF /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-
↪broadwell/intel-19.0.5.281/netcdf-fortran-4.5.2-
↪iiqxmna2ejlv4c6g5oot4t4l5wicuau2
setenv PNETCDF /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-
↪broadwell/intel-19.0.5.281/parallel-netcdf-1.12.1-
↪l5uhojc7zgamdijbbrcznb7hoygov32m
setenv WRFIO_NCD_LARGE_FILE_SUPPORT 1
setenv WRF_DFI_RADAR 1
```

2.12.2 目录结构

```
xinzhang@DESKTOP-D0HP1CH:/mnt/d/ubuntu/longrun/rap.v4.0.12$ tree -d -L 2
.
   ecf
       dump
       gempak
       pcyc
       post
       prep
   exec
   fix
   gempak
       fix
       parm
       ush
   jobs
   modulefiles
       RAP
   parm
       wmo
   scripts
   sms
       post
       wrfbufr
   sorc
       logs
       rap_full_cycle_surface.fd
       rap_gsi.fd
       rap_process_cloud.fd
       rap_process_enkf.fd
       rap_process_imssnow.fd
       rap_process_lightning.fd
       rap_process_mosaic.fd
       rap_process_sst.fd
       rap_smartinit_ak.fd
       rap_smartinit_conus.fd
       rap_smartinit_hi.fd
       rap_smartinit_pr.fd
       rap_sndp.fd
       rap_stnmlist.fd
```

(下页继续)

(续上页)

```
      rap_subflds_g2.fd
      rap_update_bc.fd
      rap_update_fields.fd
      rap_update_gvf.fd
      rap_wps.fd
      rap_wrfarw.fd
      rap_wrfbufr.fd
      rap_wrfpost.fd
  ush

46 directories
```

2.12.3 安装步骤

公用环境变量

注解:

```
setenv BASE ${ROOT_DIR}/nwprod/rap.v4.0.12/sorc
```

编译 rap_wrfarw_serial

编译步骤:

```
cd ${BASE}/rap_wrfarw.fd/WRFV3.8.1
./clean -aa
./clean -a
./clean
cd external/io_int/
rm -f *.f *.o *.obj *.i *.mod *.a diffwrf io_int_idx_tags.h
cd -
cp configure.wrf.serial configure.wrf
./compile em_real |& tee make.log
```

编译 rap_wrfpost

注解： 我们使用 *community* 版本的 UPP 下载 ：

```
cd ${BASE}
# 解压缩在该目录下
cd UPPV4.0.1
./clean -a
./configure
mkdir -p include lib bin

./compile |& tee make.log
cp bin/unipost.exe ../../exec/rap_wrfpost
```

编译 rap_update_bc

编译步骤：

```
cd ${BASE}/rap_update_bc.fd
make clean
make
```

编译 rap_wps

编译步骤：

```
cd ${BASE}/rap_wps.fd/WPSV3.9.0.1
./clean -aa
./clean -a
./clean
cp configure.wps.optim configure.wps
./compile |& tee make.log
cp ungrib/src/ungrib.exe ${BASE}/../exec/rap_wps_ungrib
cp metgrid/src/metgrid.exe ${BASE}/../exec/rap_wps_metgrid
```

注解： ${BASE}/../parm/rap_Vtable.gfs 必须替换成为 WRF-WPS 提供的给 **community** 用户使用的 Vtable

编译 rap_wrfarw

编译步骤:

```
cd ${BASE}/rap_wrfarw.fd/WRFV3.8.1
./clean -aa
./clean -a
./clean
cd external/io_int/
rm -f *.f *.o *.obj *.i *.mod *.a diffwrf io_int_idx_tags.h
cd -
cp configure.wrf.optim configure.wrf
./compile em_real |& tee make.log
cp main/real.exe ${BASE}/../exec/rap_wrfarw_real
cp main/wrf.exe ${BASE}/../exec/rap_wrfarw_fcst
```

编译 rap_gsi

编译步骤:

```
setenv NETCDF_INCLUDE "-I${NETCDF}/include"
setenv NETCDF_LDFLAGS_F "-L${NETCDF}/lib -lnetcdff -lnetcdf"
cd ${BASE}/rap_gsi.fd
cp Makefile.conf.wcoss Makefile.conf
make clean
cd gsdcloud
cp Makefile.conf.wcoss Makefile.conf
make clean
make
cd ..
make
make install
make library
```

编译 rap_process_enkf

编译步骤:

```
cd ${BASE}/rap_process_enkf.fd
cp Makefile.conf_intel Makefile.conf
make clean
```

(下页继续)

(续上页)

```
make
# it will fail, you have to manually build two files
mpif90 -FI -D_REAL8_ -DWRF -DRRR_CLOUDANALYSIS -I./ -I/glade/work/xinzhang/
↪repository/nwprod/rap.v4.0.12/sorc/rap_gsi.fd -I/glade/work/xinzhang/
↪repository/nwprod/lib/crtm/v2.2.3/intel/include/crtm_v2.2.3/ -I/glade/work/
↪xinzhang/repository/nwprod/lib/sfcio/v1.0.0/intel/include/sfcio_v1.0.0_4 -I/
↪glade/work/xinzhang/repository/nwprod/lib/sigio/v2.0.1/src/sigio_v2.0.1/
↪include -I/glade/work/xinzhang/repository/nwprod/lib/nemsio/incmod/nemsio -I/
↪glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.0.5.
↪281/netcdf-fortran-4.5.2-iiqxmna2ejlv4c6g5oot4t4l5wicuau2/include -I/glade/
↪work/xinzhang/repository/nwprod/lib/w3emc/v2.2.0/src/intel/w3emc_v2.2.0_d -
↪O3 -traceback -convert big_endian -assume byterecl -g -fp-model source -
↪qopenmp   -c dlarnv.f90
mpif90 -FI -D_REAL8_ -DWRF -DRRR_CLOUDANALYSIS -I./ -I/glade/work/xinzhang/
↪repository/nwprod/rap.v4.0.12/sorc/rap_gsi.fd -I/glade/work/xinzhang/
↪repository/nwprod/lib/crtm/v2.2.3/intel/include/crtm_v2.2.3/ -I/glade/work/
↪xinzhang/repository/nwprod/lib/sfcio/v1.0.0/intel/include/sfcio_v1.0.0_4 -I/
↪glade/work/xinzhang/repository/nwprod/lib/sigio/v2.0.1/src/sigio_v2.0.1/
↪include -I/glade/work/xinzhang/repository/nwprod/lib/nemsio/incmod/nemsio -I/
↪glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.0.5.
↪281/netcdf-fortran-4.5.2-iiqxmna2ejlv4c6g5oot4t4l5wicuau2/include -I/glade/
↪work/xinzhang/repository/nwprod/lib/w3emc/v2.2.0/src/intel/w3emc_v2.2.0_d -
↪O3 -traceback -convert big_endian -assume byterecl -g -fp-model source -
↪qopenmp   -c dlaruv.f90
make
make install
```

编译 rap_full_cycle_surface

编译步骤:

```
cd ${BASE}/rap_full_cycle_surface.fd
make clean
make
```

编译 rap_process_cloud

编译步骤:

```
cd ${BASE}/rap_process_cloud.fd
make clean
make
```

编译 rap_update_fields

编译步骤:

```
cd ${BASE}/rap_update_fields.fd
make clean
make
```

编译 rap_process_sst

编译步骤:

```
cd ${BASE}/rap_process_sst.fd
make clean
make
```

编译 rap_update_gvf

编译步骤:

```
cd ${BASE}/rap_update_gvf.fd
make clean
make
```

编译 rap_process_imssnow.fd

编译步骤:

```
cd ${BASE}/rap_process_imssnow.fd
make clean
make
```

2.13 安装 hrrr

2.13.1 公用环境变量

将如下环境变量加入到 ~/.bashrc 或 ~/.cshrc 或 ~/.profile 中

```
setenv NETCDF /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-
↪broadwell/intel-19.0.5.281/netcdf-fortran-4.5.2-
↪iiqxmna2ejlv4c6g5oot4t4l5wicuau2
setenv PNETCDF /glade/u/home/xinzhang/spack/opt/spack/linux-sles12-
↪broadwell/intel-19.0.5.281/parallel-netcdf-1.12.1-
↪l5uhojc7zgamdijbbrcznb7hoygov32m
setenv WRFIO_NCD_LARGE_FILE_SUPPORT 1
setenv WRF_DFI_RADAR 1
```

2.13.2 目录结构

```
xinzhang@DESKTOP-D0HP1CH:/mnt/e/rap_hrrr_201912/hrrr.v3.0.8$ tree -d -L 2
.
   ecf
      post
      prep
   fix
      alaska
      conus
   jobs
   modulefiles
      HRRR
   parm
      alaska
      conus
   scripts
      alaska
      conus
   sms
      post
      prep
   sorc
      hrrr_full_cycle_surface.fd
      hrrr_gsi.fd
```

(下页继续)

(续上页)

```
      hrrr_process_cloud.fd
      hrrr_process_enkf.fd
      hrrr_process_imssnow.fd
      hrrr_process_lightning.fd
      hrrr_process_mosaic.fd
      hrrr_process_sst.fd
      hrrr_ref2tten.fd
      hrrr_smartinit_ak.fd
      hrrr_smartinit_conus.fd
      hrrr_sndp.fd
      hrrr_stnmlist.fd
      hrrr_update_bc.fd
      hrrr_update_gvf.fd
      hrrr_wps.fd
      hrrr_wrfarw.fd
      hrrr_wrfbufr_alaska.fd
      hrrr_wrfbufr_conus.fd
      hrrr_wrfpost.fd
  ush
      alaska
      conus

42 directories
```

2.13.3 安装步骤

环境变量的设置

注解:

```
setenv BASE ${ROOT_DIR}/nwprod/hrrr.v3.0.8/sorc
```

编译 hrrr_wrfarw_serial

编译步骤:

```
cd ${BASE}/hrrr_wrfarw.fd/WRFV3.8.1
./clean -aa
./clean -a
./clean
cd external/io_int/
rm -f *.f *.o *.obj *.i *.mod *.a diffwrf io_int_idx_tags.h
cd -
cp configure.wrf.serial configure.wrf
./compile em_real |& tee make.log

hrrr_wrfarw.fd/WRFV3.8.1> ls -al main/*.exe
-rwxr-xr-x 1 xinzhang ncar 33662368 Apr 13 17:35 main/ndown.exe
-rwxr-xr-x 1 xinzhang ncar 34206272 Apr 13 17:35 main/real.exe
-rwxr-xr-x 1 xinzhang ncar 33077696 Apr 13 17:35 main/tc.exe
-rwxr-xr-x 1 xinzhang ncar 39798832 Apr 13 17:33 main/wrf.exe
```

编译 hrrr_update_bc

编译步骤:

```
cd ${BASE}/hrrr_update_bc.fd
make clean
make
```

编译 hrrr_wps

编译步骤:

```
cd ${BASE}/hrrr_wps.fd/WPSV3.9.0.1
./clean -aa
./clean -a
./clean
cp configure.wps.optim configure.wps
./compile |& tee make.log
cp ungrib/src/ungrib.exe ${BASE}/../exec/hrrr_wps_ungrib
cp metgrid/src/metgrid.exe ${BASE}/../exec/hrrr_wps_metgrid
```

编译 hrrr_wrfpost

注解: Please use the *community* upp

cp /${ROOT_DIR}/nwprod/rap.v4.0.12/exec/rap_wrfpost ${BASE}/../exec/hrrr_wrfpost

编译 hrrr_wrfarw

编译步骤:

```
cd ${BASE}/hrrr_wrfarw.fd/WRFV3.8.1
./clean -aa
./clean -a
./clean
cd external/io_int/
rm -f *.f *.o *.obj *.i *.mod *.a diffwrf io_int_idx_tags.h
cd -
cp configure.wrf.optim configure.wrf
./compile em_real |& tee make.log
cp main/real.exe ${BASE}/../exec/hrrr_wrfarw_real
cp main/wrf.exe ${BASE}/../exec/hrrr_wrfarw_fcst
```

编译 hrrr_gsi

编译步骤:

```
setenv NETCDF_INCLUDE "-I${NETCDF}/include"
setenv NETCDF_LDFLAGS_F "-L${NETCDF}/lib -lnetcdff -lnetcdf"
setenv CRTM_INC /glade/work/xinzhang/repository/nwprod/lib/crtm/v2.2.3/intel/
↪include/crtm_v2.2.3
setenv CRTM_LIB /glade/work/xinzhang/repository/nwprod/lib/crtm/v2.2.3/intel/
↪libcrtm_v2.2.3.a
cd ${BASE}/hrrr_gsi.fd
make clean
cd gsdcloud
make clean
make
cd ..
make
make install
make library
```

编译 hrrr_process_enkf

编译步骤:

```
cd ${BASE}/hrrr_process_enkf.fd
make clean
make
mpif90 -FI -D_REAL8_ -DWRF -DRRR_CLOUDANALYSIS -I./ -I/glade/work/xinzhang/
↪repository/nwprod/hrrr.v3.0.8/sorc/hrrr_gsi.fd -I/glade/work/xinzhang/
↪repository/nwprod/lib/crtm/v2.2.3/intel/include/crtm_v2.2.3 -I/glade/work/
↪xinzhang/repository/nwprod/lib/sfcio/v1.0.0/intel/include/sfcio_v1.0.0_4 -I/
↪glade/work/xinzhang/repository/nwprod/lib/sigio/v2.0.1/src/sigio_v2.0.1/
↪include -I/glade/work/xinzhang/repository/nwprod/lib/nemsio/incmod/nemsio -I/
↪glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.0.5.
↪281/netcdf-fortran-4.5.2-iiqxmna2ejlv4c6g5oot4t4l5wicuau2/include -I/glade/
↪work/xinzhang/repository/nwprod/lib/w3emc/v2.2.0/src/intel/w3emc_v2.2.0_d -
↪O3 -traceback -convert big_endian -assume byterecl -g -fp-model source -
↪qopenmp   -c dlarnv.f90
mpif90 -FI -D_REAL8_ -DWRF -DRRR_CLOUDANALYSIS -I./ -I/glade/work/xinzhang/
↪repository/nwprod/hrrr.v3.0.8/sorc/hrrr_gsi.fd -I/glade/work/xinzhang/
↪repository/nwprod/lib/crtm/v2.2.3/intel/include/crtm_v2.2.3 -I/glade/work/
↪xinzhang/repository/nwprod/lib/sfcio/v1.0.0/intel/include/sfcio_v1.0.0_4 -I/
↪glade/work/xinzhang/repository/nwprod/lib/sigio/v2.0.1/src/sigio_v2.0.1/
↪include -I/glade/work/xinzhang/repository/nwprod/lib/nemsio/incmod/nemsio -I/
↪glade/u/home/xinzhang/spack/opt/spack/linux-sles12-broadwell/intel-19.0.5.
↪281/netcdf-fortran-4.5.2-iiqxmna2ejlv4c6g5oot4t4l5wicuau2/include -I/glade/
↪work/xinzhang/repository/nwprod/lib/w3emc/v2.2.0/src/intel/w3emc_v2.2.0_d -
↪O3 -traceback -convert big_endian -assume byterecl -g -fp-model source -
↪qopenmp   -c dlaruv.f90
make
make install
```

编译 hrrr_full_cycle_surface

编译步骤:

```
cd ${BASE}/hrrr_full_cycle_surface.fd
make clean
make
```

编译 hrrr_process_cloud

编译步骤:

```
cd ${BASE}/hrrr_process_cloud.fd
make clean
make
```

编译 hrrr_process_imssnow

编译步骤:

```
cd ${BASE}/hrrr_process_imssnow.fd
make clean
make
```

编译 hrrr_process_mosaic

编译步骤:

```
cd ${BASE}/hrrr_process_mosaic.fd
make clean
make
```

编译 hrrr_process_sst

编译步骤:

```
cd ${BASE}/hrrr_process_sst.fd
make clean
make
```

编译 hrrr_process_lightning

编译步骤:

```
cd ${BASE}/hrrr_process_lightning.fd
make clean
make
```

编译 hrrr_update_gvf

编译步骤:

```
cd ${BASE}/hrrr_update_gvf.fd
make clean
make
```

编译 hrrr_ref2tten

编译步骤:

```
cd ${BASE}/hrrr_ref2tten.fd
make clean
make
```

CHAPTER 3

操作手册

3.1 时间标识的建立

- 整个 RAP-HRRR 系统的运行是依靠 `${ROOT_DIR}/com/date` 下的时间标识文件来告诉系统运行哪个时刻.

```
mkdir -p ${ROOT_DIR}/com/date
```

- 在定时作业中加入如下的脚本，每个整点更新时间标识文件:

```
crontab -e
# For RAP
# ----------------

# Touch the time hourly
# ---------------------------

0 * * * * . $HOME/.profile ; ${ROOT_DIR}/utils/chinawise_touchtime.py
```

3.2 ecFlow workflow

3.2.1 修改 ecf_kill, ecf_status 和 ecf_submit

ecflow 使用 ${ROOT_DIR}/workflow 下的 ecf_kill, ecf_status 和 ecf_submit 进行作业的中止，查询和提交，所以需要用户根据所使用的 HPC 上的作业管理系统做出相应的修改。尤其是 hostname, 需要和 ECF_HOST 一致。

3.2.2 修改 head.h

在 ${ROOT_DIR}/workflow/include/head.h 中定义了一些全局变量，用户可以根据需要进行修改，例如:

```
# Tell ecFlow we have started
export NCARG_ROOT=/cma/u/app/ncl-6.1.2

# For ecflow
export PATH=%ROOT_DIR%/ecflow/sharelib/bin:$PATH
export LIBPATH=${NCARG_ROOT}/lib:/opt/freeware/lib64:/cma/u/app/lib::/cma/u/app/netcdf/
↪lib:/cma/u/app/udunits/lib:/sya/u/gongying/LongRun/sharelib

# Common area
export jobid=$job_name.$$
export SENDDBN=NO
export KEEPDATA=NO
export ROOT_DIR=%ROOT_DIR%
export GESROOT=%ROOT_DIR%/nwges
export NWROOT=%ROOT_DIR%/nwprod
export WGRIB2=$NWROOT/grib_util.v1.1.1/exec/wgrib2
export CRTMDIR=%ROOT_DIR%/nwprod/rap.v4.0.12/sorc/UPPV4.0.1/src/lib/crtm2/src/fix
export NDATE=$NWROOT/prod_util.v1.1.3/ush/ndate
export PATH=$NWROOT/prod_util.v1.1.3/ush:$PATH

# RAP
export COMGFS=%ROOT_DIR%/com/gfs/prod
export COMINgfs=$COMGFS
export COMROOTp1=$GESROOT

# HRRR
export dom=conus
export COMROOT=%ROOT_DIR%/com
```

(下页继续)

(续上页)

```
export COMRAP=%ROOT_DIR%/com/rap/prod
export DATAROOT=%ROOT_DIR%/tmpnwprd1
export utilscript=%ROOT_DIR%/nwprod/util/ush
export jlogfile=%ROOT_DIR%/com/logs/jlogfiles/jlogfile.${jobid}
export FIXcrtm_post=${CRTMDIR}
```

3.2.3 RAP 作业集的生成

以下是如何产生 RAP 的作业集供 ecFlow 调用:

```
> cd ${ROOT_DIR}/workflow
> rm -fr sy_nwp_rap_v2020.def
> ./sy_nwp_rap_v2020.py
```

其产生的 sy_nwp_rap_v2020.def 为整个作业集的 ecFlow 描述

3.2.4 加载作业集

```
> ecflow_client --load=sy_nwp_rap_v2020.def
> ecflow_client --suites
sy_nwp_rap_v2020
```

加载后打开 ecflow_ui , 界面类似如下:

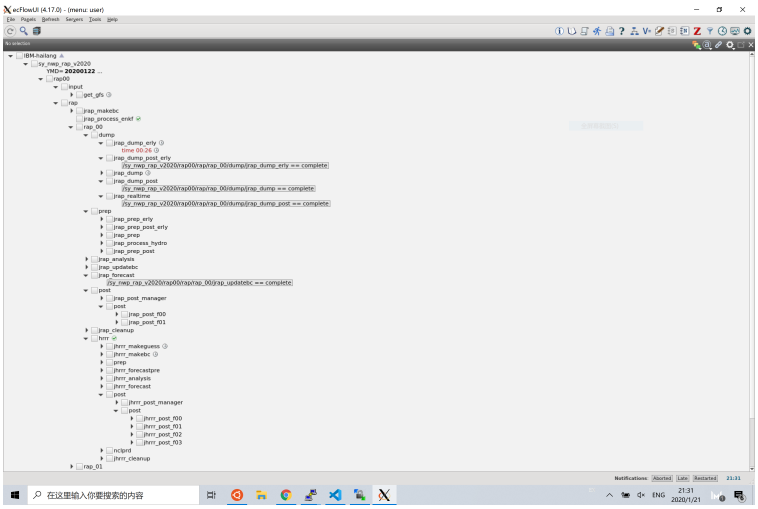


图 1: ecflowv_ui 加载截图

3.2.5 启动作业集

```
> ecflow_client --begin=sy_nwp_rap_v2020
```

启动后打开 ecflow_ui，界面类似如下：

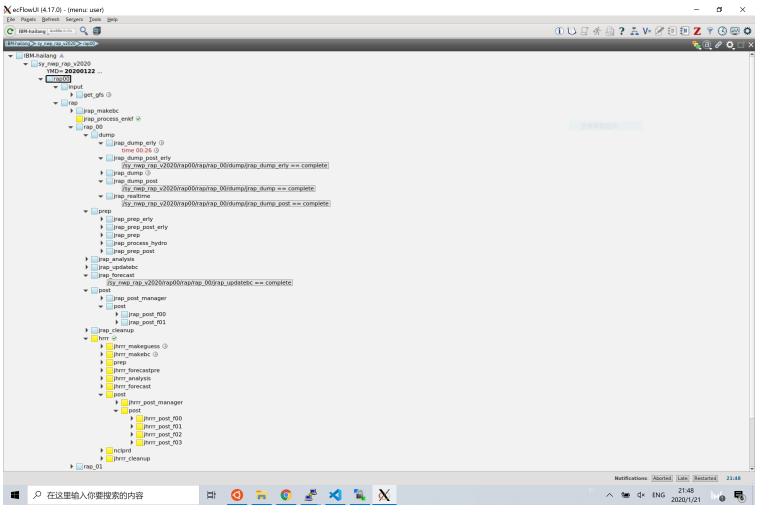


图 2: ecflowv_ui 启动截图

3.2.6 重新更换已有的作业集

```
> ecflow_client --suites
sy_nwp_rap_v2020

> ecflow_client --replace=/sy_nwp_rap sy_nwp_rap_v201908.def false force
```

3.3 err_exit

根据所使用的作业管理系统修改：${ROOT_DIR}/nwprod/prod_util.v1.1.3/ush/err_exit

例如使用 PBS 系统：

```
...
...
if [ -n "$PBS_JOBID" ]; then
  if [ "$SENDECF" = "YES" ]; then
    timeout 30 ecflow_client --msg "$ECF_NAME: $msg1"
  fi
```

(下页继续)

(续上页)

```
  qdel $PBS_JOBID
  sleep 60  # Wait for PBS to kill the job
fi
```

3.4 输出文件目录的准备

- 建立日志文件输出目录:

```
mkdir -p ${ROOT_DIR}/com/output/prod/today
mkdir -p ${ROOT_DIR}/com/logs/jlogfiles
mkdir -p ${ROOT_DIR}/nwges/prod/hrrr/hrrrges_sfc/conus
```

3.5 GFS 资料的准备

注解: 沈阳大气所的实时的 0.5 度的 GFS 资料由气象台负责下载，每天 4 个时次：00Z, 06Z, 12Z, 18Z; 资料位置 /sya/g3/wrf/data/gfs/ccyy; 间隔 6 小时，一直到 108 小时。

为了可以在 RAP 系统中使用，必须做以下操作:

```
cd ${ROOT_DIR}/com
mkdir -p gfs/prod
cd gfs/prod
ln -fs /sya/g3/wrf/data/gfs/2020 2020
```

3.6 海温资料的准备

- 实时的 0.083 度的海温资料应该每天实时下载，每天下载一次，资料在每天 22:34Z 可以下载，资料位置在:

```
https://ftp.ncep.noaa.gov/data/nccf/com/gfs/prod/sst.20200209/rtgssthr_grb_
↪0.083.grib2
```

- 为了可以在 RAP (cyc = 01), HRRR (cyc = 02) 中使用，必须做以下操作:

```
$ cd workflow/include
$ edit head.h
# Modify export COMINsst=/path/of/downloaded/sst/file

# 请修改如下脚本，设置正确的 SST 路径：
$ ~/rap/nwprod/rap.${rap_ver}/scripts/exrap_check_sst.sh.ecf
$ ~/rap/nwprod/rap.${rap_ver}/scripts/exrap_makebc.sh.ecf
```

- `rap_makebc` 会使用 *SST*
- `rap_process_sst` 的作用主要是修改初始场中的 *SST* 和 *TSK* ，特别是水体，冰面的温度。以下是一个列子，显示更新之前后的 *SST* 的对比：

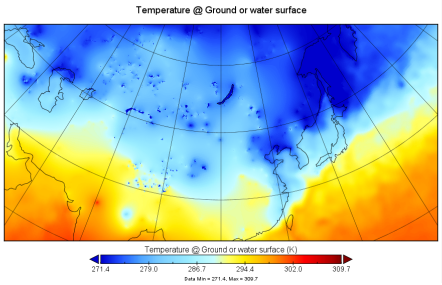


图 3: global SST analysis @ 2020-02-21

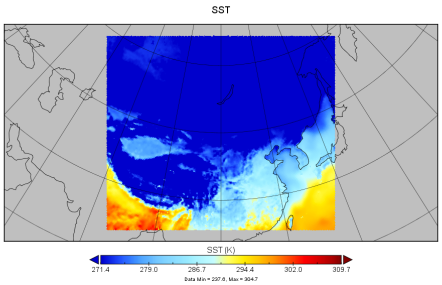


图 4: before *rap_process_sst* @ 2020-02-21

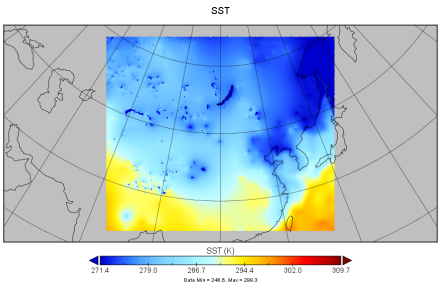


图 5: after *rap_process_sst* @ 2020-02-21

- 以下是一个列子，显示更新之前后的 *TSK* 的对比：

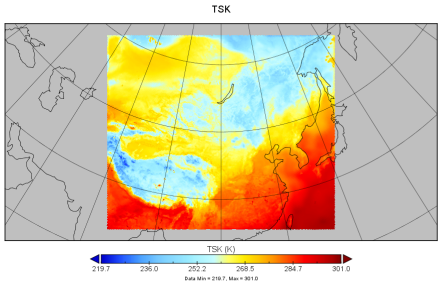


图 6: before *rap_process_sst* @ 2020-02-21

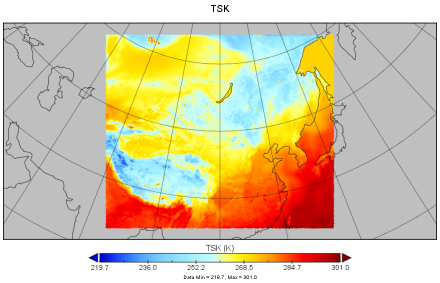


图 7: after *rap_process_sst* @ 2020-02-21

3.7 冰雪资料的准备

- 实时的北半球 1 公里分辨率冰雪资料，每天下载一次，资料位置在：

```
https://www.natice.noaa.gov/ims/grib2_v3_p2_2020_4km.html
```

- 为了可以在 *RAP* 和 *HRRR* 系统中使用，必须做以下操作:

```
# in cron jobs, please do sync as following
15,30,45,0 * * * * rsync -av /path/of/imssnow/rap.20* ${ROOT_DIR}/com/rap/
↪prod
# Populate to *rap_e* directory for *HRRR* access
15,30,45,0 * * * * ${ROOT_DIR}/utils/imssnow_populate.py


# 请修改如下脚本，设置正确的 SNOWICE 数据的路径：
$  ${ROOT_DIR}/nwprod/rap.v4.0.12/scripts/exrap_check_imssnow.sh.ecf
$  ${ROOT_DIR}/nwprod/rap.v4.0.12/scripts/exrap_analysis.sh.ecf
```

- *rap* (cyc = 00)

```
exrap_analysis.sh.ecf
# process_imssnow run after DA
```

- *hrrr* (cyc = 00)

```
exhrrr_analysis.sh.ecf
#  process_imssnow run after DA
```

- `process_imssnow` 修改如下变量:

1. LU_INDEX
2. SEAICE
3. IVGTYP
4. ISLTYP
5. SNOW
6. SNOWH
7. SNOWC
8. XLAND
9. LANDMASK

3.8 绿植覆盖率资料的准备

- 实时的周平均全球绿植覆盖率资料，每天下载一次，资料位置在:

```
https://satepsanone.nesdis.noaa.gov/pub/VHP/GVF/GVF-WKL-GLB_v2r3_npp_
↪s20200201_e20200207_c202002081017160.grib2
```

- set *GVF* path in `workflow/include/head.h`

```
export GVF==%ROOT_DIR%/com/gvf
```

- *rap* `exrap_analysis_pcyc.sh.ecf`

  在 cyc = 03 and 15 运行

- *hrrr* `exhrrr_prep_ref2tten.sh.ecf`

  每个 cycle 运行

- `rap_update_gvf` 的作用，是修改初始场中的 *VEGFRA* 。

以下是一个列子，显示更新之前后的 *VEGFRAC* 的对比：

先给出 EU 的观测：

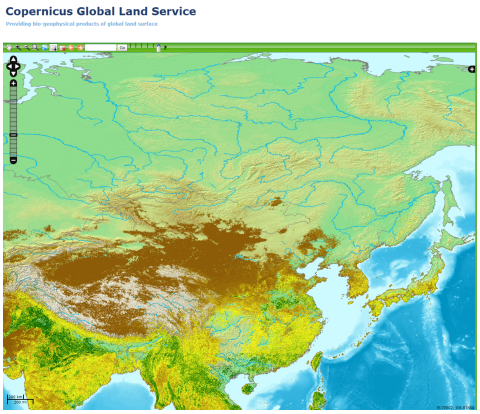


图 8: EU observation @ 2020-02-22

下面是更新前后的对比：

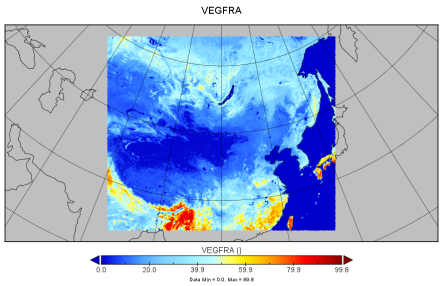


图 9: before *rap_update_gvf* @ 2020-02-22

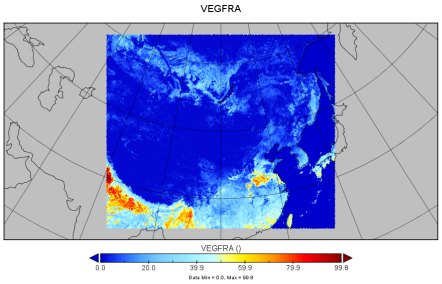


图 10: after *rap_update_gvf* @ 2020-02-22

- 3KM 区域例子

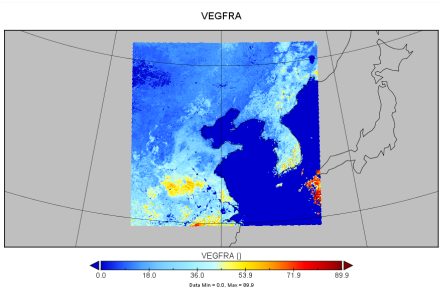


图 11: before *hrrr_update_gvf* @ 2020-02-22

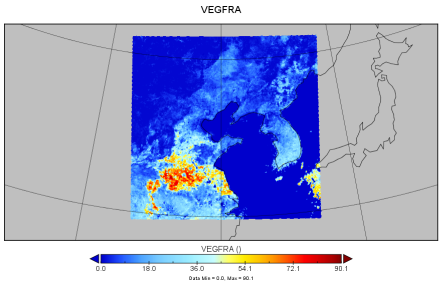


图 12: after *hrrr_update_gvf* @ 2020-02-22

3.9 磁盘空间的清理

因为 RAP 和 HRRR 系统在运行期间会产生大量的临时数据和产品，为了防止磁盘空间不足，可以在定时作业中加入以下定时作业，下面例子为每小时 00 分，删除 1 天前的所有数据文件。所以，请用户自行做好数据备份的工作

在定时作业中加入:

```
$ crontab -l

# Remove old files
# ------------------------

0 * * * * find /sya/g2/hailang/chinawise/com/rap/prod/ -type f ! -name "*.png" -mtime +1␣
↪-exec rm -f {} +
0 * * * * find /sya/g2/hailang/chinawise/com/hrrr/prod/ -type f ! -name "*.png" -mtime␣
↪+1 -exec rm -f {} +
0 * * * * find /sya/g2/hailang/chinawise/com/output/prod/today/ -type f  -mtime +1 -exec␣
↪rm -f {} +
0 * * * * find /sya/g2/hailang/chinawise/com/logs/jlogfiles/ -type f  -mtime +1 -exec rm␣
↪-f {} +
0 * * * * find /sya/g2/hailang/chinawise/nwges/prod/ -type f -mmin +720 -exec rm -f {} +
```

(下页继续)

(续上页)

```
0 * * * * find /sya/g2/hailang/chinawise/tmpnwprd1 -type d -mmin +360 -exec rm -fr {} +
```

3.10 观测资料的解码

3.10.1 TANK 目录的准备

```
mkdir -p ${ROOT_DIR}/dcom/us007003
```

在 ${ROOT_DIR}/dcom/us007003 中，各类资料的存放目录参见表格 DataInventory

3.10.2 ax file (GTS)

Data directory structure

The ax files are organized every 60 minutes :

```
[hailang@sya01n01 ~/chinawise/nwprod/decoders]$ls -la /sya/g3/wrf/data/obs/2001*.ax
 ......
-rw-r--r-- 1 wrf work 4183866 Jan 26 03:32 /sya/g3/wrf/data/obs/20012512.ax
-rw-r--r-- 1 wrf work 1566223 Jan 26 03:32 /sya/g3/wrf/data/obs/20012513.ax
-rw-r--r-- 1 wrf work 1583806 Jan 26 03:32 /sya/g3/wrf/data/obs/20012514.ax
-rw-r--r-- 1 wrf work 2342846 Jan 26 03:32 /sya/g3/wrf/data/obs/20012515.ax
-rw-r--r-- 1 wrf work 1535829 Jan 26 03:32 /sya/g3/wrf/data/obs/20012516.ax
-rw-r--r-- 1 wrf work 1540626 Jan 26 03:32 /sya/g3/wrf/data/obs/20012517.ax
-rw-r--r-- 1 wrf work 3959697 Jan 26 03:32 /sya/g3/wrf/data/obs/20012518.ax
-rw-r--r-- 1 wrf work 1466181 Jan 26 03:32 /sya/g3/wrf/data/obs/20012519.ax
-rw-r--r-- 1 wrf work 1464897 Jan 26 03:31 /sya/g3/wrf/data/obs/20012520.ax
-rw-r--r-- 1 wrf work 2309281 Jan 26 03:31 /sya/g3/wrf/data/obs/20012521.ax
-rw-r--r-- 1 wrf work 1555754 Jan 26 03:31 /sya/g3/wrf/data/obs/20012522.ax
-rw-r--r-- 1 wrf work 1470354 Jan 26 03:31 /sya/g3/wrf/data/obs/20012523.ax
-rw-r--r-- 1 wrf work 3975865 Jan 26 03:31 /sya/g3/wrf/data/obs/20012600.ax
-rw-r--r-- 1 wrf work 1655233 Jan 26 03:31 /sya/g3/wrf/data/obs/20012601.ax
-rw-r--r-- 1 wrf work 1430912 Jan 26 03:31 /sya/g3/wrf/data/obs/20012602.ax
-rw-r--r-- 1 wrf work  849702 Jan 26 03:31 /sya/g3/wrf/data/obs/20012603.ax
```

Data format

样例文件 2020-01-26_00Z 报文

```
ZCZC 734

SAKB31 NGTT 260000

METAR NGTA 260000Z 04003KT 9999 FEW018TCU 32/26 Q1008 RMK TCU TO NE=




NNNN

ZCZC 741

SASO31 AGGG 260000

METAR 260000

METAR AGGH 260000Z 34006KT 9999 2CU020 31/25 Q1007=




NNNN

ZCZC 860

SAAU69 AMMC 260000

METAR YSTW 260000Z AUTO 32005KT 9999 // SCT061 28/22 Q1015=

METAR YSWG 260000Z AUTO 29005KT 230V340 9999 // NCD 29/05 Q1013=

METAR YTEF 260000Z AUTO 13013KT 9999 // NCD 33/08 Q1010=

METAR YTNK 260000Z AUTO 15013KT 9999 // NCD 29/19 Q1011=

METAR YTRE 260000Z AUTO 07005KT 9999 // NCD 31/21 Q1012=

METAR YWGT 260000Z AUTO 04005KT 360V120 9999 // NCD 27/09 Q1013=
```

(下页继续)

(续上页)

```
METAR YWHA 260000Z AUTO 14010KT 9999 // BKN017 22/16 Q1016=

METAR YWLM 260000Z AUTO 34007KT 9999 // NCD 32/20 Q1012=




NNNN

ZCZC 861

SAAU62 AMMC 260000

METAR YBTL 260000Z AUTO 22004KT 190V250 9999 -SHRA OVC096 26/25 Q1010=

METAR YBUD 260000Z AUTO 07011KT 9999 // SCT033 SCT041 BKN049 29/20 Q1013=

METAR YBWP 260000Z AUTO 30007KT 9999 // SCT022 SCT029 BKN032 32/26 Q1010=

METAR YBWX 260000Z AUTO 25007KT //// // ////// 29/24 Q1008=

METAR YCAR 260000Z AUTO 20015KT 9999 // NCD 26/21 Q1009=

METAR YCAS 260000Z AUTO 05010KT //// // ////// 28/21 Q1015=

METAR YCBA 260000Z AUTO 19009KT 9999 // NCD 32/12 Q1012=

METAR YCBB 260000Z AUTO 36012KT 9999 // SCT021 BKN032 26/20 Q1016=

METAR YCBP 260000Z AUTO 13011KT 9999 // NCD 30/// Q1015=

METAR YCCY 260000Z AUTO 11006KT 9999 // BKN008 28/24 Q1009=




NNNN

ZCZC 862
```

(下页继续)

(续上页)

```
SASO31 AGGG 260000

METAR 260000

METAR AGGM 260000Z 00000KT 9999 FEW018 31/25 Q1008=




NNNN

ZCZC 863

SANC01 NWWW 260000

METAR NWWW 260000Z AUTO 14009KT 060V190 9999 BKN054 31/18 Q1010

TEMPO 4000 SHRA FEW030TCU=
```

Data extractor

Decoder source code

1. Source code directory:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimisssurf/sorc
```

2. Subroutines to decode surface data

   - `lsdcod.f`

注解:

- The *cimiss.tbl* and `decod_WMO.Res40.headers` are not used, although they are required as arguments and read in.
- The path and file name of `cimiss_data_surface` file are hard coded in the subroutines.

---

4. Compile the code

```
> make
```

Decode and convert to BUFR format

1. enter into the exec directory

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimisssurf/exec
> ls -la
total 2028
drwxr-xr-x. 3 zwtd zwtd     211 Dec 20 22:45 .
drwxr-xr-x. 6 zwtd zwtd      82 Dec 20 22:42 ..
lrwxrwxrwx. 1 zwtd zwtd      34 Dec  9 23:48 bufrtab.000 -> ../../decod_shared/fix/
↪bufrtab.000
-rw-r--r--. 1 zwtd zwtd 1121683 Dec  9 23:48 cimiss.tbl
lrwxrwxrwx. 1 zwtd zwtd      31 Dec  9 23:48 decod_WMO.Res40.headers -> ../parm/decod_
↪WMO.Res40.headers
-rwxr-xr-x. 1 zwtd zwtd  940696 Dec 20 00:24 decod_dccimiss
-rw-r--r--. 1 zwtd zwtd     462 Dec 20 22:45 decod_dccimiss.log
-rwxr-xr-x. 1 zwtd zwtd     484 Dec 20 00:30 run.ksh
-rwxr-xr-x. 1 zwtd zwtd     776 Dec 20 22:44 run_dccimisssurf.py
drwxr-xr-x. 2 zwtd zwtd      58 Dec 20 22:45 tmp
```

2. we provide a script to run the decoder in batch mode:

```
> ./run_dccimisssurf.py -s 2018121600 -e 2018121700 -i 1
```

注解:

- given the starting datetime and ending datetime, it iterates all cycles (every 1 hours)
- the units of interval is hour (-i)
- this script call run.ksh

---

3. run the decoder script

```
> run.ksh

> cat run.ksh
!/bin/bash
export DBNBUFRT=120
export TRANJB=/g5/nwpc_ep8/obsproc/nwprod/ush/tranjb
export tank_dir=/g5/nwpc_ep8/obsproc/nwprod/dcom/us007003
export DBNROOT=`pwd`
rm tmp/*
```

(下页继续)

(续上页)

```
rm decod_dccimiss.log
./decod_dccimiss -d decod_dccimiss.log -b 240 -c $1 bufrtab.000 cimiss.tbl decod_WMO.
↪Res40.headers
ls -la tmp/*

BUFR_FILES=$(echo tmp/BUFR*)
echo ${BUFR_FILES}

for file in ${BUFR_FILES}
do
  ${TRANJB} ${tank_dir} ${file}
done
```

注解:

- -c $1 : Set the **current time** (201809011600) used to calculate the time departures of the obs. data.
- -b 240 : Number of hours to decode prior to "current" time (default)
- The observations with date/time between **current time** - 240 hours and **current time** + 3 are **kept**.

4. The generated BUFR format file will be saved at

```
> ls -la tmp
-rw-r--r--  1 xinzhang  staff  4199744 Sep 21 18:45 tmp/BUFR.0.cimiss.1.6436.1536097072.8
```

Transfer bufr data to BUFR Tanks

- put data in BUFR **tanks**:

```
> /g5/nwpc_ep8/obsproc/nwprod/ush/tranjb /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003 tmp/
↪BUFR.0.cimiss.1.6436.1536097072.8

> ls -al /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/20180901/b000/xx01
-rw-r--r-- 1 vagrant vagrant 4235328 Sep 27 04:42 /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20180901/b000/xx001
```

注解:

- **if environmental variable SCREEN=ON :**

  - Define **Run Time** is the system time when the tranjb is running.
  - Only observations with date/time between **Run Time** - 10 days and **Run Time** + 12 hours are kept.

- for retrospective run, set **SCREEN=OFF**
- `/g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/yyyymmdd/bmmm/xxsss` (where mmm is WMO BUFR message type and xxx is local BUFR message subtype)
- 000.001 (in dump group mnemonic adpsfc): Surface synoptic fixed land reports
- BUFR format
- Arranged by UTC day and continuously grow throughout the day, if you run decoders many time, the content of the file will grow
- No QC (other than rudimentary checks inside decoders)
- No duplicate checking
- Interested users can use utility `debufr` to check the content of the bufr file:

```
> /g5/nwpc_ep8/obsproc/nwprod/util/exec/debufr /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20180901/b000/xx001
```

  the output is in `debufr.out`.

3.10.3 `ta` file (GTS)

Data directory structure

The `ta` files are organized every 60 minutes :

```
[hailang@sya01n01 ~/chinawise/nwprod/decoders]$ls -la /sya/g3/wrf/data/obs/2001*.ta
 ......
-rw-r--r-- 1 wrf work 4183866 Jan 26 03:32 /sya/g3/wrf/data/obs/20012512.ta
-rw-r--r-- 1 wrf work 1566223 Jan 26 03:32 /sya/g3/wrf/data/obs/20012513.ta
-rw-r--r-- 1 wrf work 1583806 Jan 26 03:32 /sya/g3/wrf/data/obs/20012514.ta
-rw-r--r-- 1 wrf work 2342846 Jan 26 03:32 /sya/g3/wrf/data/obs/20012515.ta
-rw-r--r-- 1 wrf work 1535829 Jan 26 03:32 /sya/g3/wrf/data/obs/20012516.ta
-rw-r--r-- 1 wrf work 1540626 Jan 26 03:32 /sya/g3/wrf/data/obs/20012517.ta
-rw-r--r-- 1 wrf work 3959697 Jan 26 03:32 /sya/g3/wrf/data/obs/20012518.ta
-rw-r--r-- 1 wrf work 1466181 Jan 26 03:32 /sya/g3/wrf/data/obs/20012519.ta
-rw-r--r-- 1 wrf work 1464897 Jan 26 03:31 /sya/g3/wrf/data/obs/20012520.ta
-rw-r--r-- 1 wrf work 2309281 Jan 26 03:31 /sya/g3/wrf/data/obs/20012521.ta
```

(下页继续)

(续上页)

```
-rw-r--r-- 1 wrf work 1555754 Jan 26 03:31 /sya/g3/wrf/data/obs/20012522.ta
-rw-r--r-- 1 wrf work 1470354 Jan 26 03:31 /sya/g3/wrf/data/obs/20012523.ta
-rw-r--r-- 1 wrf work 3975865 Jan 26 03:31 /sya/g3/wrf/data/obs/20012600.ta
-rw-r--r-- 1 wrf work 1655233 Jan 26 03:31 /sya/g3/wrf/data/obs/20012601.ta
-rw-r--r-- 1 wrf work 1430912 Jan 26 03:31 /sya/g3/wrf/data/obs/20012602.ta
-rw-r--r-- 1 wrf work  849702 Jan 26 03:31 /sya/g3/wrf/data/obs/20012603.ta
```

Data format

样例文件 2020-01-26_00Z 报文

```
ZCZC 101

UAFJ01 NFFN 260000

AIREP

ANZ99 2441S16626E 2304 F360 MS48 000/024=

FJI911 2633S16503E 2304 F410 MS60 215/016=

JST4 0615S17931E 2306 F400 MS53 236/007=

ANZ99 2334S16540E 2313 F375 MS52 355/010=

ANZ77 2929S16534E 2317 F380 MS52 311/019=

KAL130 2700S16252E 2317 F340 MS43 248/007=

ANZ99 2306S16518E 2318 F380 MS53 326/005=

FJI911 2736S16323E 2318 F410 MS59 199/018=

JST4 0754S17838E 2320 F400 MS52 002/013=

ANZ77 2902S16505E 2321 F380 MS53 192/028=

ANZ77 2806S16403E 2330 F380 MS52 309/007=
```

(下页继续)

(续上页)

```
ANZ99 2134S16409E 2332 F380 MS52 276/020=

JST4 0924S17725E 2334 F400 MS53 319/020=

CES779 1147S16040E 2336 F370 MS46 039/018=

JST4 1021S17638E 2342 F400 MS52 024/019=

QFA4 0045N17551W 2344 F360 MS44 107/015=

ANZ77 2644S16234E 2344 F380 MS53 327/008=

ANZ99 2004S16303E 2345 F380 MS52 324/015=

QFA4 0026N17611W 2347 F376 MS48 139/018=

QFA4 0024N17613W 2347 F377 MS48 137/021=

JST4 1054S17610E 2347 F400 MS51 351/031=

QFA4 0018N17620W 2349 F380 MS50 134/019=

CES779 1320S16144E 2350 F370 MS48 320/012=

CES779 1338S16156E 2352 F370 MS47 196/010=

ASY446 2249S16049E 2353 F290 MS30 005/010=

CES779 1359S16211E 2355 F370 MS48 355/006=

CES779 1401S16212E 2356 F370 MS48 354/007=

QFA4 0031S17711W 2358 F380 MS50 112/018=

ANZ77 3055S16706E 2303 F380 MS53 029/029=

KAL130 2824S16423E 2303 F340 MS44 191/017=
```

(下页继续)

(续上页)

```
NNNN

ZCZC 102

UAAE01 WSSS 260000

AXM5108 0415N 10935E 2353 F330 MS37 175/17KT=




NNNN

ZCZC 104

UANT01 CWAO 260000

ARP DAL132 5112N05306W 0000 F370 5200N05000W 0015 5300N04000W MS55

 297/72 KT

N823NW DDL YAY 260000 J41A




NNNN

ZCZC 106

UANT01 CWAO 260000
```

Data extractor

Decoder source code

1. Source code directory:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimisssurf/sorc
```

2. Subroutines to decode surface data

   - `lsdcod.f`

注解:

- The *cimiss.tbl* and `decod_WMO.Res40.headers` are not used, although they are required as arguments and read in.
- The path and file name of `cimiss_data_surface` file are hard coded in the subroutines.

---

4. Compile the code

```
> make
```

Decode and convert to BUFR format

1. enter into the exec directory

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimisssurf/exec
> ls -la
total 2028
drwxr-xr-x. 3 zwtd zwtd     211 Dec 20 22:45 .
drwxr-xr-x. 6 zwtd zwtd      82 Dec 20 22:42 ..
lrwxrwxrwx. 1 zwtd zwtd      34 Dec  9 23:48 bufrtab.000 -> ../../decod_shared/fix/
↪bufrtab.000
-rw-r--r--. 1 zwtd zwtd 1121683 Dec  9 23:48 cimiss.tbl
lrwxrwxrwx. 1 zwtd zwtd      31 Dec  9 23:48 decod_WMO.Res40.headers -> ../parm/decod_
↪WMO.Res40.headers
-rwxr-xr-x. 1 zwtd zwtd  940696 Dec 20 00:24 decod_dccimiss
-rw-r--r--. 1 zwtd zwtd     462 Dec 20 22:45 decod_dccimiss.log
-rwxr-xr-x. 1 zwtd zwtd     484 Dec 20 00:30 run.ksh
-rwxr-xr-x. 1 zwtd zwtd     776 Dec 20 22:44 run_dccimisssurf.py
drwxr-xr-x. 2 zwtd zwtd      58 Dec 20 22:45 tmp
```

2. we provide a script to run the decoder in batch mode:

```
> ./run_dccimisssurf.py -s 2018121600 -e 2018121700 -i 1
```

注解:

- given the starting datetime and ending datetime, it iterates all cycles (every 1 hours)
- the units of interval is hour (-i)
- this script call run.ksh

3. run the decoder script

```
> run.ksh

> cat run.ksh
!/bin/bash
export DBNBUFRT=120
export TRANJB=/g5/nwpc_ep8/obsproc/nwprod/ush/tranjb
export tank_dir=/g5/nwpc_ep8/obsproc/nwprod/dcom/us007003
export DBNROOT=`pwd`
rm tmp/*
rm decod_dccimiss.log
./decod_dccimiss -d decod_dccimiss.log -b 240 -c $1 bufrtab.000 cimiss.tbl decod_WMO.
↪Res40.headers
ls -la tmp/*

BUFR_FILES=$(echo tmp/BUFR*)
echo ${BUFR_FILES}

for file in ${BUFR_FILES}
do
  ${TRANJB} ${tank_dir} ${file}
done
```

注解:

- -c $1 : Set the **current time** (201809011600) used to calculate the time departures of the obs. data.
- -b 240 : Number of hours to decode prior to "current" time (default)
- The observations with date/time between **current time** - 240 hours and **current time** + 3 are **kept**.

4. The generated BUFR format file will be saved at

```
> ls -la tmp
-rw-r--r--  1 xinzhang  staff  4199744 Sep 21 18:45 tmp/BUFR.0.cimiss.1.6436.1536097072.8
```

Transfer bufr data to BUFR Tanks

- put data in BUFR **tanks**:

```
> /g5/nwpc_ep8/obsproc/nwprod/ush/tranjb /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003 tmp/
↪BUFR.0.cimiss.1.6436.1536097072.8

> ls -al /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/20180901/b000/xx01
-rw-r--r-- 1 vagrant vagrant 4235328 Sep 27 04:42 /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20180901/b000/xx001
```

注解:

- **if environmental variable SCREEN=ON :**
  - Define **Run Time** is the system time when the tranjb is running.
  - Only observations with date/time between **Run Time** - 10 days and **Run Time** + 12 hours are kept.
- for retrospective run, set **SCREEN=OFF**
- `/g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/yyyymmdd/bmmm/xxsss` (where mmm is WMO BUFR message type and xxx is local BUFR message subtype)
- 000.001 (in dump group mnemonic adpsfc): Surface synoptic fixed land reports
- BUFR format
- Arranged by UTC day and continuously grow throughout the day, if you run decoders many time, the content of the file will grow
- No QC (other than rudimentary checks inside decoders)
- No duplicate checking
- Interested users can use utility `debufr` to check the content of the bufr file:

  ```
  > /g5/nwpc_ep8/obsproc/nwprod/util/exec/debufr /g5/nwpc_ep8/obsproc/nwprod/dcom/
  ↪us007003/20180901/b000/xx001
  ```

  the output is in `debufr.out`.

3.10.4 Aircraft

Data directory structure

The Aircraft data is organized as CIMISS files every 6 hours :

```
> ls -la /g3/tianwh/share/conobs/rec_RSING_*:
-rw-r--r-- 1 tianwh obs 34351730 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060100_g_06.dat
-rw-r--r-- 1 tianwh obs 16639350 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060106_g_06.dat
-rw-r--r-- 1 tianwh obs 25511010 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060112_g_06.dat
-rw-r--r-- 1 tianwh obs 31514910 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060118_g_06.dat
-rw-r--r-- 1 tianwh obs 29063650 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060200_g_06.dat
-rw-r--r-- 1 tianwh obs 14376670 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060206_g_06.dat
-rw-r--r-- 1 tianwh obs 24181990 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060212_g_06.dat
-rw-r--r-- 1 tianwh obs 32740190 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060218_g_06.dat
-rw-r--r-- 1 tianwh obs 32471950 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060300_g_06.dat
-rw-r--r-- 1 tianwh obs 20091890 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060306_g_06.dat
-rw-r--r-- 1 tianwh obs 25002950 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060312_g_06.dat
-rw-r--r-- 1 tianwh obs 32896430 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060318_g_06.dat
-rw-r--r-- 1 tianwh obs 34783490 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060400_g_06.dat
-rw-r--r-- 1 tianwh obs 21610470 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060406_g_06.dat
-rw-r--r-- 1 tianwh obs 29883770 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060412_g_06.dat
-rw-r--r-- 1 tianwh obs 34280610 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060418_g_06.dat
-rw-r--r-- 1 tianwh obs 35147210 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSING_
↪2019060500_g_06.dat
```

Data format

CIMISS format data is looks like:

```
    RSING    2019       6       1       0  245369       5       4       4
OOHQVXZA     44.49    150.37       1.00    241.00       9.00
  11247.00    -43.95    222.00      38.60       9.00       9.00       9.00       9.00
OOHQVXZA     46.01    152.68       1.00    221.00       9.00
  11217.00    -47.95    216.00      38.60       9.00       9.00       9.00       9.00
OOHQVXZA     47.45    155.04       1.00    201.00       9.00
  10638.00    -48.95    229.00      45.30       9.00       9.00       9.00       9.00
OOHQVXZA     53.18    167.60       1.00     37.00       3.00
  10368.00    -48.45    267.00      13.40       9.00       9.00       9.00       9.00
OOHQVXZA     56.68    178.56      31.00   2340.00       3.00
  10368.00    -50.45    345.00      44.20       9.00       9.00       9.00       9.00
OOHQVXZA     49.34    158.56       1.00    135.00       3.00
  10368.00    -47.45    232.00      25.20       9.00       9.00       9.00       9.00
OOHQVXZA     58.23   -172.53      31.00   2301.00       3.00
  10367.00    -49.95      7.00      11.80       9.00       9.00       9.00       9.00
OOHQVXZA     51.98    164.39       1.00     56.00       3.00
  10366.00    -50.15    207.00      23.10       9.00       9.00       9.00       9.00
OOHQVXZA     52.59    165.94       1.00     47.00       3.00
  10366.00    -49.45    227.00      20.60       9.00       9.00       9.00       9.00
OOHQVXZA     50.69    161.37       1.00    116.00       3.00
  10366.00    -47.95    274.00      25.70       9.00       9.00       9.00       9.00
OOHQVXZA     50.02    159.94       1.00    125.00       3.00
  10364.00    -47.65    249.00      24.20       9.00       9.00       9.00       9.00
OOHQVXZA     57.15   -179.26      31.00   2330.00       3.00
  10364.00    -46.95    358.00      29.80       9.00       9.00       9.00       9.00
OOHQVXZA     54.93    171.98       1.00     12.00       9.00
  10363.00    -48.95    320.00      32.90       9.00       9.00       9.00       9.00
OOHQVXZA     59.00   -160.01      31.00   2212.00       9.00
  10363.00    -44.95     83.00       6.20       9.00       9.00       9.00       9.00
OOHQVXZA     53.75    169.23       1.00     28.00       9.00
  10363.00    -48.95    287.00      18.50       9.00       9.00       9.00       9.00
OOHQVXZA     57.55   -177.04      31.00   2320.00       3.00
  10362.00    -47.95      1.00      20.60       9.00       9.00       9.00       9.00
OOHQVXZA     51.34    162.86       1.00    106.00       3.00
  10362.00    -47.65    259.00      14.40       9.00       9.00       9.00       9.00
OOHQVXZA     54.28    170.43       1.00     21.00       9.00
  10333.00    -47.95    314.00      23.10       9.00       9.00       9.00       9.00
```

(下页继续)

(续上页)

```
00HQVXZA     55.56    174.12      1.00      1.00      9.00
  10333.00    -50.95    328.00     34.00      9.00      9.00      9.00      9.00
00HQVXZA     58.77   -167.53     31.00   2241.00      9.00
  10333.00    -50.95     39.00     12.90      9.00      9.00      9.00      9.00
00HQVXZA     57.53   -177.16     31.00   2321.00      9.00
  10333.00    -46.95      5.00     21.10      9.00      9.00      9.00      9.00
00HQVXZA     50.32    160.57      1.00    121.00      9.00
  10333.00    -46.95    261.00     21.60      9.00      9.00      9.00      9.00
```

Data extractor

A python code is used to extract the desired information from this CIMISS file:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders
> rm cimiss_amdar_data
> ./read_nmc_cimiss_rsing.py -f /g3/tianwh/share/conobs/rec_RSING_2019060100_g_06.dat
```

If you want to batch process number of CIMISS files, you can use following command:

```
> # This command will find all CIMISS files and prcessing the file one by one
> decod_nmc_cimiss_rsing.py -d /g3/tianwh/share/conobs
```

The information we want to extract from CIMISS is:

```
import fortranformat as ff
output_line = ff.FortranRecordWriter('(A8, 5A8, 4f10.2, I8, f10.2)')
output.write(output_line.write([call_name, obstime.strftime("%Y"), obstime.strftime("%m
↪"), \
        obstime.strftime("%d"), obstime.strftime("%H"), obstime.strftime("%M"), lat, lon,
↪ phi, \
        T, dd, ff]) + "\n")
```

the content of `micaps_amdar_data` is:

```
> less micaps_amdar_data
01ZXL5JA    2019      06      30      14      38     34.61   -109.08  11884.00    -56.45␣
↪    277      7.20
01ZXL5JA    2019      06      30      14      32     33.93   -109.51  11883.00    -55.25␣
↪    277      7.70
01ZXL5JA    2019      06      30      14      50     35.94   -108.19  11882.00    -56.95␣
↪    295      9.30
```

(下页继续)

(续上页)

```
01ZXL5JA    2019    06    30    14    59    37.01    -107.83    11882.00    -57.15␣
↪    282    11.80
01ZXL5JA    2019    06    30    14    56    36.64    -107.90    11882.00    -56.95␣
↪    280    10.30
01ZXL5JA    2019    06    30    14    41    34.94    -108.86    11881.00    -56.25␣
↪    273    9.80
01ZXL5JA    2019    06    30    14    47    35.61    -108.41    11881.00    -56.45␣
↪    282    9.30
01ZXL5JA    2019    06    30    14    44    35.28    -108.64    11881.00    -56.45␣
↪    278    9.30
01ZXL5JA    2019    06    30    14    53    36.28    -107.99    11881.00    -56.65␣
↪    286    9.80
01ZXL5JA    2019    06    30    14    35    34.27    -109.29    11880.00    -56.45␣
↪    273    6.70
01ZXL5JA    2019    06    30    14    29    33.59    -109.72    11468.00    -52.75␣
↪    287    6.70
01ZXL5JA    2019    06    30    14    26    33.26    -109.94    9933.00    -39.75␣
↪    158    1.00
01ZXL5JA    2019    06    30    14    23    32.90    -110.16    8393.00    -27.25␣
↪    232    3.10
01ZXL5JA    2019    06    30    14    19    32.55    -110.38    6529.00    -12.65␣
↪    265    0.50
01ZXL5JA    2019    06    30    14    19    32.51    -110.40    6358.00    -11.75␣
↪    305    1.50
01ZXL5JA    2019    06    30    14    19    32.48    -110.42    6135.00    -10.45␣
↪    310    1.50
01ZXL5JA    2019    06    30    14    18    32.45    -110.44    5902.00    -8.75␣
↪    334    1.50
01ZXL5JA    2019    06    30    14    18    32.42    -110.46    5637.00    -7.45␣
↪    336    3.10
01ZXL5JA    2019    06    30    14    18    32.39    -110.47    5379.00    -5.75␣
↪    340    2.10
01ZXL5JA    2019    06    30    14    17    32.35    -110.49    5122.00    -3.75␣
↪    15    2.10
01ZXL5JA    2019    06    30    14    17    32.32    -110.51    4757.00    -0.75␣
↪    47    3.10
01ZXL5JA    2019    06    30    14    17    32.29    -110.53    4485.00    1.55␣
↪    34    1.50
01ZXL5JA    2019    06    30    14    16    32.26    -110.55    4214.00    3.55␣
↪    24    3.10
```

(下页继续)

(续上页)

```
01ZXL5JA    2019    06    30    14    16    32.24    -110.58    3946.00    5.85␣
↪    31    3.10
01ZXL5JA    2019    06    30    14    16    32.22    -110.62    3665.00    8.85␣
↪    34    3.60
01ZXL5JA    2019    06    30    14    15    32.21    -110.65    3390.00    11.05␣
↪    17    3.60
01ZXL5JA    2019    06    30    14    15    32.19    -110.68    3270.00    11.55␣
↪    28    1.50
01ZXL5JA    2019    06    30    14    15    32.17    -110.71    3111.00    13.85␣
↪    65    2.10
01ZXL5JA    2019    06    30    14    14    32.15    -110.73    2809.00    16.35␣
↪    157    1.50
01ZXL5JA    2019    06    30    14    14    32.14    -110.76    2511.00    18.85␣
↪    126    4.60
01ZXL5JA    2019    06    30    14    14    32.12    -110.79    2160.00    22.05␣
↪    131    4.10
01ZXL5JA    2019    06    30    14    13    32.11    -110.81    1942.00    23.55␣
↪    131    3.10
01ZXL5JA    2019    06    30    14    13    32.10    -110.83    1797.00    24.25␣
↪    103    2.10
01ZXL5JA    2019    06    30    14    13    32.08    -110.85    1606.00    25.85␣
↪    108    2.60
01ZXL5JA    2019    06    30    14    13    32.08    -110.86    1576.00    26.05␣
↪    110    3.10
01ZXL5JA    2019    06    30    14    13    32.08    -110.86    1527.00    26.35␣
↪    111    3.60
01ZXL5JA    2019    06    30    14    12    32.08    -110.87    1487.00    27.05␣
↪    117    3.60
01ZXL5JA    2019    06    30    14    12    32.08    -110.88    1445.00    27.35␣
↪    127    4.10
01ZXL5JA    2019    06    30    14    12    32.08    -110.88    1397.00    27.55␣
↪    133    4.60
01ZXL5JA    2019    06    30    14    12    32.08    -110.89    1351.00    28.05␣
↪    132    4.60
01ZXL5JA    2019    06    30    14    12    32.08    -110.89    1305.00    28.35␣
↪    124    4.10
01ZXL5JA    2019    06    30    14    12    32.08    -110.90    1260.00    28.35␣
↪    109    4.10
01ZXL5JA    2019    06    30    14    12    32.09    -110.90    1220.00    28.85␣
↪    108    4.10
```

(下页继续)

(续上页)

```
01ZXL5JA    2019      06      30      14      12      32.09    -110.91    1171.00      29.25␣
↪     103      3.60
01ZXL5JA    2019      06      30      14      12      32.09    -110.91    1104.00      29.55␣
↪     103      3.10
01ZXL5JA    2019      06      30      14      12      32.10    -110.92    1023.00      28.85␣
↪     119      2.10
01ZXL5JA    2019      06      30      14      11      32.10    -110.92     944.00      28.85␣
↪     153      1.00
01ZXL5JA    2019      06      30      14      11      32.10    -110.92     866.00      29.25␣
↪     104      0.00
01ZXL5JA    2019      06      30      14      11      32.11    -110.93     802.00      31.55␣
↪       0      0.00
020UFPRA    2019      06      30      14      53      35.03     -96.72   11582.00     -52.15␣
↪      24      6.70
020UFPRA    2019      06      30      14      38      35.03     -94.32   11552.00     -52.55␣
↪      30     14.40
020UFPRA    2019      06      30      14      23      35.01     -91.94   11552.00     -53.05␣
↪      66      2.10
020UFPRA    2019      06      30      10      45      30.08     -83.79   11552.00     -53.75␣
↪     233     12.90
020UFPRA    2019      06      30      14      08      34.09     -89.61   10973.00     -50.05␣
↪     107      2.60
020UFPRA    2019      06      30      13      53      34.04     -87.33   10942.00     -49.55␣
↪     161      6.70
020UFPRA    2019      06      30      10      30      29.02     -82.58   10668.00     -46.25␣
↪     249      4.60
```

Decoder source code

1. Source code directory:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dcmicapsamdar/sorc
```

2. Subroutines to decode Aircraft data

   - `afdcod.f`

注解:

- The `pirep.tbl`, `airep.tbl` are not used, although they are required as arguments and read in.
- The path and file name of `cmicaps_amdar_data` file are hard coded in the subroutines.

4. Compile the code

```
> make
```

Decode and convert to BUFR format

1. enter into the exec directory

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dcmicapsamdar/exec
> ls -la
total 1944
drwxr-xr-x  9 xinzhang  staff     288 Sep 25 21:24 .
drwxr-xr-x  5 xinzhang  staff     160 Sep 21 18:45 ..
lrwxr-xr-x  1 xinzhang  staff      25 Sep 21 18:45 airep.tbl -> ../dictionaries/airep.tbl
lrwxr-xr-x  1 xinzhang  staff      34 Sep 21 18:45 bufrtab.004 -> ../../decod_shared/fix/
↪bufrtab.004
-rwxr-xr-x  1 xinzhang  staff  985984 Sep 21 18:45 decod_dcmicapsamdar
lrwxr-xr-x  1 xinzhang  staff      25 Sep 21 18:45 pirep.tbl -> ../dictionaries/pirep.tbl
-rwxr-xr-x  1 xinzhang  staff     410 Sep 21 18:45 run.ksh
drwxr-xr-x  3 xinzhang  staff      96 Sep 21 18:45 tmp
```

2. we provide a script to run the decoder in batch mode:

```
> ./run_dcmicapsamdar.py -s 2018121612 -e 2018121700 -i 1
```

注解:

- given the starting datetime and ending datetime, it iterates all cycles (every 1 hours)
- the units of interval is hour (-i)
- this script call run.ksh

3. run the decoder script

```
> run.ksh

> cat run.ksh
#!/bin/bash
export DBNBUFRT=120
export TRANJB=/g5/nwpc_ep8/obsproc/nwprod/ush/tranjb
```

(下页继续)

(续上页)

```
export tank_dir=/g5/nwpc_ep8/obsproc/nwprod/dcom/us007003
export SCREEN="OFF"
export DBNROOT=`pwd`
rm tmp/*
rm decod_dcmicapsadmar.log
./decod_dcmicapsadmar -v 4 -d decod_dcmicapsadmar.log  -b 240 -c $1 pirep.tbl airep.tbl␣
↪bufrtab.004
ls -la tmp/*

BUFR_FILES=$(echo tmp/BUFR*)
echo ${BUFR_FILES}

for file in ${BUFR_FILES}
do
  ${TRANJB} ${tank_dir} ${file}
done
```

注解:

- -c $1 : Set the **current time** (201809120200) used to calculate the time departures of the obs. data.
- -b 240 : Number of hours to decode prior to "current" time (default)
- The observations with date/time between **current time** - 240 hours and **current time** + 3 are **kept**.

---

4. The generated BUFR format file will be saved at

```
> ls -la tmp/BUFR.0.aircraft.1.1933.1537419287.73
-rw-r--r--  1 xinzhang  staff  1806552 Sep 21 18:45 tmp/BUFR.0.aircraft.1.1933.
↪1537419287.73
```

Transfer bufr data to BUFR Tanks

- put data in BUFR **tanks**:

```
> /g5/nwpc_ep8/obsproc/nwprod/ush/tranjb /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003 tmp/
↪BUFR.0.aircraft.1.1933.1537419287.73

> ls -al /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/20180912/b004/xx003
-rw-r--r-- 1 vagrant vagrant 1828720 Sep 19 22:54 /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20180912/b004/xx003
```

注解:

- **if environmental variable SCREEN=ON :**
  - Define **Run Time** is the system time when the tranjb is running.
  - Only observations with date/time between **Run Time** - 10 days and **Run Time** + 12 hours are kept.
- for retrospective run, set **SCREEN=OFF**
- /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/yyyymmdd/bmmm/xxsss (where mmm is WMO BUFR message type and xxx is local BUFR message subtype)
- 004.003 (in dump group mnemonic aircft): AMDAR format aircraft data from ASDAR/ACARS reporting systems
- BUFR format
- Arranged by UTC day and continuously grow throughout the day, if you run decoders many time, the content of the file will grow
- No QC (other than rudimentary checks inside decoders)
- No duplicate checking
- Interested users can use utility *debufr* to check the content of the bufr file:

```
> /g5/nwpc_ep8/obsproc/nwprod/util/exec/debufr /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20180912/b004/xx003
```

  the output is in `debufr.out`.

3.10.5 Radiosonde

Data directory structure

The radiosonde data is organized as CIMISS files every 6 hours :

```
>  ls -la /g3/tianwh/share/conobs/rec_RTEMP_20190*
-rw-r--r-- 1 tianwh obs  5573840 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019072906_g_06.dat
-rw-r--r-- 1 tianwh obs 35183202 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019072912_g_06.dat
-rw-r--r-- 1 tianwh obs  3148125 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019072918_g_06.dat
```

(下页继续)

(续上页)

```
-rw-r--r-- 1 tianwh obs 40329560 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019073000_g_06.dat
-rw-r--r-- 1 tianwh obs  4675228 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019073006_g_06.dat
-rw-r--r-- 1 tianwh obs 38562886 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019073012_g_06.dat
-rw-r--r-- 1 tianwh obs  3265073 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019073018_g_06.dat
-rw-r--r-- 1 tianwh obs 39276942 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019073100_g_06.dat
-rw-r--r-- 1 tianwh obs 19319961 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019073106_g_06.dat
-rw-r--r-- 1 tianwh obs 49400604 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019073112_g_06.dat
-rw-r--r-- 1 tianwh obs  9822680 9 月  25 01:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019073118_g_06.dat
-rw-r--r-- 1 tianwh obs 54987251 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081000_g_06.dat
-rw-r--r-- 1 tianwh obs  9648840 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081006_g_06.dat
-rw-r--r-- 1 tianwh obs 48108084 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081012_g_06.dat
-rw-r--r-- 1 tianwh obs 11045732 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081018_g_06.dat
-rw-r--r-- 1 tianwh obs 54517933 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081100_g_06.dat
-rw-r--r-- 1 tianwh obs 11997517 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081106_g_06.dat
-rw-r--r-- 1 tianwh obs 46280155 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081112_g_06.dat
-rw-r--r-- 1 tianwh obs  9347275 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081118_g_06.dat
-rw-r--r-- 1 tianwh obs 59498350 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081200_g_06.dat
-rw-r--r-- 1 tianwh obs 13860934 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081206_g_06.dat
-rw-r--r-- 1 tianwh obs 47197672 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081212_g_06.dat
-rw-r--r-- 1 tianwh obs 11938607 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081218_g_06.dat
```

(下页继续)

(续上页)

```
-rw-r--r-- 1 tianwh obs 60678158 8 月  14 06:13 /g3/tianwh/share/conobs/rec_RTEMP_
↪2019081300_g_06.dat
......
```

Data format

CIMISS radiosonde format data is looks like:

```
RTEMP    2019       8      13      12     730       6       7       6       3
    1001     70.94     -8.67      9.00     13.00   1112.00  2890
   1013.50 131072.00      9.00      7.80      4.03     81.00     10.20      0.00      0.
↪00      0.00      1.00      0.00      0.00      0.00      0.00      0.00
   1012.30   2048.00     19.00      7.96      3.41     39.00      4.00      2.00     -0.
↪00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1011.20      0.00     27.00      7.85      3.31     28.00      4.10      4.00     -0.
↪00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1009.90      0.00     39.00      7.74      3.24     34.00      4.80      6.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1008.80   2048.00     48.00      7.66      3.23     36.00      5.30      8.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1007.60      0.00     57.00      7.59      3.28     38.00      5.70     10.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1006.50      0.00     65.00      7.50      3.20     39.00      6.10     12.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1005.20      0.00     75.00      7.42      3.03     41.00      6.40     14.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1003.80      0.00     87.00      7.37      2.93     42.00      6.70     16.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1002.30      0.00     98.00      7.23      2.97     44.00      6.80     18.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1000.70      0.00    113.00      7.12      2.95     46.00      6.90     20.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   1000.00  65536.00    119.00      7.05      2.94     48.00      7.00     21.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
    999.00      0.00    127.00      6.96      2.93     48.00      7.00     22.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
    997.40      0.00    141.00      6.83      2.82     51.00      6.90     24.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
    995.70      0.00    154.00      6.71      2.81     53.00      6.90     26.00     -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

(下页继续)

(续上页)

```
   994.00      0.00    168.00      6.58      2.78     56.00      6.80     28.00      -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   992.40      0.00    183.00      6.44      2.72     59.00      6.70     30.00      -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   990.90      0.00    196.00      6.32      2.72     62.00      6.60     32.00      -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   989.40      0.00    207.00      6.19      2.74     65.00      6.50     34.00      -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   988.10      0.00    218.00      6.02      2.83     68.00      6.40     36.00      -0.
↪00     -0.00      0.00      0.00      0.00      0.00      0.00      0.00
   986.80      0.00    228.00      5.92      2.86     71.00      6.30     38.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   985.50      0.00    238.00      5.82      2.81     73.00      6.30     40.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   984.30      0.00    250.00      5.71      2.79     75.00      6.20     42.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   983.00   2048.00    261.00      5.59      2.80     77.00      6.10     44.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   981.70      0.00    271.00      5.49      2.78     78.00      6.00     46.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   980.40      0.00    281.00      5.40      2.81     78.00      6.00     48.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   979.00      0.00    293.00      5.29      2.73     79.00      5.90     50.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   977.60      0.00    304.00      5.20      2.71     79.00      5.90     52.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   976.20      0.00    316.00      4.98      2.77     79.00      5.80     54.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   974.70      0.00    329.00      4.91      2.61     78.00      5.80     56.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   973.20      0.00    344.00      4.86      2.45     78.00      5.80     58.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   971.70      0.00    354.00      4.76      2.44     78.00      5.80     60.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   970.30      0.00    366.00      4.63      2.42     77.00      5.80     62.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   969.00      0.00    377.00      4.51      2.41     77.00      5.90     64.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   967.70      0.00    387.00      4.42      2.41     77.00      6.00     66.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
```

(下页继续)

(续上页)

```
   966.40      0.00    399.00      4.29      2.51     77.00      6.10     68.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   965.10      0.00    411.00      4.17      2.49     77.00      6.20     70.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   963.90      0.00    420.00      4.06      2.53     77.00      6.30     72.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   962.70      0.00    431.00      3.94      2.51     77.00      6.30     74.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   961.60      0.00    441.00      3.84      2.40     78.00      6.40     76.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   960.70      0.00    449.00      3.80      2.29     78.00      6.40     78.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   960.10      0.00    454.00      3.83      2.11     79.00      6.30     80.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   959.50      0.00    457.00      3.89      2.07     80.00      6.20     82.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   959.00      0.00    461.00      3.88      2.10     80.00      6.00     84.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   958.30      0.00    465.00      3.82      2.08     81.00      5.80     86.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   957.50      0.00    472.00      3.75      2.09     82.00      5.50     88.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   956.60      0.00    481.00      3.66      2.11     82.00      5.20     90.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   955.60      0.00    490.00      3.56      2.08     82.00      4.80     92.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   954.60      0.00    499.00      3.46      1.98     81.00      4.40     94.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   953.70      0.00    508.00      3.37      1.95     79.00      4.00     96.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   952.80      0.00    513.00      3.30      1.98     76.00      3.70     98.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   952.00      0.00    521.00      3.24      2.01     72.00      3.40    100.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   951.10      0.00    530.00      3.11      2.01     66.00      3.20    102.00      -0.
↪00     -0.01      0.00      0.00      0.00      0.00      0.00      0.00
   950.30      0.00    536.00      3.06      1.91     59.00      3.00    104.00      -0.
↪00     -0.02      0.00      0.00      0.00      0.00      0.00      0.00
   949.50      0.00    542.00      3.02      1.90     51.00      3.00    106.00      -0.
↪00     -0.02      0.00      0.00      0.00      0.00      0.00      0.00
```

(下页继续)

(续上页)

```
   948.80      0.00    549.00      2.94      1.93     43.00      3.00    108.00     -0.
↪00     -0.02      0.00      0.00      0.00      0.00      0.00      0.00
   948.00      0.00    555.00      2.86      1.83     36.00      3.10    110.00     -0.
↪00     -0.02      0.00      0.00      0.00      0.00      0.00      0.00
   947.20      0.00    560.00      2.81      1.77     30.00      3.30    112.00     -0.
↪00     -0.02      0.00      0.00      0.00      0.00      0.00      0.00
   946.40      0.00    570.00      2.73      1.75     25.00      3.50    114.00     -0.
↪00     -0.02      0.00      0.00      0.00      0.00      0.00      0.00
   945.50      0.00    576.00      2.66      1.67     21.00      3.60    116.00     -0.
↪00     -0.02      0.00      0.00      0.00      0.00      0.00      0.00
```

注解: The above sample include Mandatory levels reports and Significant levels reports, CIMISS uses *EVSS* to identify the type.

- EVSS=" 131072" : Surface level
- EVSS=" 65536" : Mandatory levels
- EVSS=" 2048" : Significant levels wind
- EVSS=" 8192" : Significant levels temperature

Data extractor

A python code is used to extract the desired information from this XML files:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders
> rm upr_data
> ./read_nmc_cimiss_rtemp.py -f  /g3/tianwh/share/conobs/rec_RTEMP_2019081300_g_06.dat
```

If you want to batch process number of CIMISS files, you can use following command:

```
> # This command will find all CIMISS files and prcessing the file one by one
> rm upr_data
> decod_nmc_cimiss_rtemp.py -d /g3/tianwh/share/conobs
```

The information we want to extract from XML is:

```
import fortranformat as ff
output_line = ff.FortranRecordWriter('(2I8, 3f10.2, 6A8, 6f10.2)')
output.write(output_line.write([id, int(rid), lat+lat_d, lon+lon_d, elv, obstime.
↪strftime("%Y"), obstime.strftime("%m"), \
```

(下页继续)

(续上页)

```
          obstime.strftime("%d"), obstime.strftime("%H"), obstime.strftime("%M"),␣
↪obstime.strftime("%S"), \
          p, phi, T, Td, dd, ff]) + "\n")
```

The content of `upr_data` is:

```
> less upr_data
56046 131072 33.7667 99.65 3968 2018 9 1 11 16 0 631 999999 10.2 8.8 0 0
56046 65536 33.7667 99.65 3968 2018 9 1 11 16 0 500 5870 0.2 -1 160 2
56046 65536 33.7667 99.65 3968 2018 9 1 11 16 0 400 7630 -9.3 -10.6 255 3
56046 65536 33.7667 99.65 3968 2018 9 1 11 16 0 300 9800 -22.9 -26.6 290 8
56046 65536 33.7667 99.65 3968 2018 9 1 11 16 0 250 11110 -32.3 -37.3 310 7
56046 65536 33.7667 99.65 3968 2018 9 1 11 16 0 200 12640 -45.7 -55.7 295 4
56046 65536 33.7667 99.65 3968 2018 9 1 11 16 0 150 14490 -62.3 999999 305 6
56046 65536 33.7667 99.65 3968 2018 9 1 11 16 0 100 16880 -80.1 999999 265 4
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 999998 999998 999998 999998 115 6
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 999998 999998 999998 999998 75 3
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 999998 999998 999998 999998 95 1
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 999998 999998 999998 999998 5 2
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 999998 999998 999998 999998 130 6
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 999998 999998 999998 999998 15 5
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 962 999999 35.2 20.2 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 767 999999 16.6 14.6 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 724 999999 15.6 9.6 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 632 999999 7.6 5.6 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 524 999999 1.6 -5.4 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 497 999999 0.8 -15.2 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 428 999999 -6.7 -21.7 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 402 999999 -8.3 -48.3 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 355 999999 -15.9 -33.9 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 312 999999 -22.5 -58.5 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 207 999999 -44.7 -74.7 999999 999999
57328 8192 31.2 107.5 344.9 2018 9 1 11 16 0 137 999999 -64.7 999999 999999 999999
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 962 999999 999999 999999 135 3
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 781 999998 999999 999999 75 3
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 751 999998 999999 999999 100 3
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 669 999998 999999 999999 85 3
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 643 999998 999999 999999 135 1
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 619 999998 999999 999999 70 1
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 595 999998 999999 999999 165 3
```

(下页继续)

(续上页)

```
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 573 999998 999999 999999 180 4
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 552 999998 999999 999999 140 2
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 532 999998 999999 999999 140 2
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 513 999998 999999 999999 100 2
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 493 999998 999999 999999 5 2
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 473 999998 999999 999999 5 1
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 452 999998 999999 999999 240 2
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 425 999998 999999 999999 190 3
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 410 999998 999999 999999 180 4
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 377 999998 999999 999999 120 6
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 345 999998 999999 999999 75 4
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 330 999998 999999 999999 40 3
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 316 999998 999999 999999 10 5
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 233 999998 999999 999999 45 16
57328 2048 31.2 107.5 344.9 2018 9 1 11 16 0 156 999998 999999 999999 35 23
```

Decoders Source code

1. Source code directory:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimissupr/sorc
```

2. Key subroutines to decode Radiosonde data

   - `uadcod_mandatory.f`
   - `uadcod_significant_temp.f`
   - `uadcod_significant_wind.f`

3. The top control program is `dccimissupr.c`, the code snippet is:

```
/*
** Call the decoding routine.
**
** Change this function call and define command for the
** specific decoder.
*/

#ifdef UNDERSCORE
#define ua_dcod_mandatory ua_dcod_mandatory_
#define ua_dcod_significant_wind ua_dcod_significant_wind_
```

(下页继续)

(续上页)

```
#define ua_dcod_significant_temp ua_dcod_significant_temp_
#endif

    ua_dcod_mandatory ( curtim, lndtbl, shptbl, bufrtb, &nhours, &iret,
              strlen(curtim), strlen(lndtbl),
              strlen(shptbl), strlen(bufrtb) );

    ua_dcod_significant_wind ( curtim, lndtbl, shptbl, bufrtb, &nhours, &iret,
              strlen(curtim), strlen(lndtbl),
              strlen(shptbl), strlen(bufrtb) );

    ua_dcod_significant_temp ( curtim, lndtbl, shptbl, bufrtb, &nhours, &iret,
              strlen(curtim), strlen(lndtbl),
              strlen(shptbl), strlen(bufrtb) );

/*
**  Send shut down message and close the log files.
*/
    dc_exit ( &iret );

}
```

注解:

- The `lndtbl`, `shptbl` are not used actually, although they are required as arguments and read in.
- The path and file name of `../../upr_data` file are hard-wired in the subroutines.

4. Compile the code

```
> make
```

Decode and convert to BUFR format

1. enter into the exec directory

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimissupr/exec
> ls -la
total 964
drwxr-xr-x 1 vagrant vagrant    288 Sep 10 16:31 .
```

(下页继续)

(续上页)

```
drwxr-xr-x 1 vagrant vagrant    160 Sep  8 21:04 ..
lrwxr-xr-x 1 vagrant vagrant     34 Aug 30 04:15 bufrtab.002 -> ../../decod_shared/fix/
↪bufrtab.002
-rw-r--r-- 1 vagrant vagrant    449 Sep 10 16:29 decod_dccimissupr.log
-rwxr-xr-x 1 vagrant vagrant 947632 Sep 10 16:19 decod_dccmissupr
-rwxr-xr-x 1 vagrant vagrant    389 Sep  9 05:40 run.ksh
lrwxr-xr-x 1 vagrant vagrant     30 Aug 30 04:15 sonde.land.tbl -> ../dictionaries/sonde.
↪land.tbl
lrwxr-xr-x 1 vagrant vagrant     30 Aug 30 04:15 sonde.ship.tbl -> ../dictionaries/sonde.
↪ship.tbl
drwxr-xr-x 1 vagrant vagrant     96 Sep 10 16:29 tmp
```

2. we provide a script to run the decoder in batch mode:

```
> ./run_dccimissupr.py -s 2018121600 -e 2018121700
```

注解:

- given the starting datetime and ending datetime, it iterates all cycles (00Z, 12Z)
- the interval is 12 hours for radiosonde
- this script call run.ksh

```
> cat run.ksh
#!/bin/bash
if [[ $# -eq 0 ]]; then
    echo "No time cycle is given"
    echo "Usage:: run.ksh 180430/1000"
    exit
fi

export DBNBUFRT=120    # control the frequency to flush the bufr
export TRANJB=/g5/nwpc_ep8/obsproc/nwprod/ush/tranjb
export tank_dir=/g5/nwpc_ep8/obsproc/nwprod/dcom/us007003    # used by TRANJB
export SCREEN="OFF"    # Turn off the time screen and keep all data, used by TRANJB
export DBNROOT=`pwd`
rm tmp/*
rm decod_dccimissupr.log
./decod_dccmissupr -d decod_dccimissupr.log -b 240 -c $1 sonde.land.tbl sonde.ship.tbl␣
↪bufrtab.002
```

(下页继续)

(续上页)

```
ls -la tmp/*
```

注解:

- -c $1 : Set the **current time** (201809011200) used to calculate the time departures of the obs. data.
- -b 240 : Number of hours to decode prior to "current" time (default)
- The observations with date/time between **current time** - 240 hours and **current time** + 3 are **kept**.

3. The generated BUFR format file will be saved at

Transfer bufr data to BUFR Tanks

- put data in BUFR **tanks**:

```
> /g5/nwpc_ep8/obsproc/nwprod/ush/tranjb /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003 tmp/
↪BUFR.0.raob.1.12381.1536602459.61

> ls -al /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/20180901/b002/xx001
-rw-r--r-- 1 vagrant vagrant 36304 Sep 10 16:29 /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20180901/b002/xx001
```

注解:

- **if environmental variable SCREEN=ON :**
  - Define **Run Time** is the system time when the tranjb is running.
  - Only observations with date/time between **Run Time** - 10 days and **Run Time** + 12 hours are kept.
- for retrospective run, set **SCREEN=OFF**
- `/nwprod/dcom/us007003/yyyymmdd/bmmm/xxsss` (where mmm is WMO BUFR message type and xxx is local BUFR message subtype)
- 002.001 (in dump group mnemonic adpupa): Fixed radiosonde land reports
- BUFR format
- Arranged by UTC day and continuously grow throughout the day, if you run decoders many time, the content of the file will grow
- No QC (other than rudimentary checks inside decoders)

- No duplicate checking
- Interested users can use utility `debufr` to check the content of the bufr file:

```
> /g5/nwpc_ep8/obsproc/nwprod/util/exec/debufr /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20180901/b002/xx001
```

the output is in `debufr.out`.

3.10.6 Surface

Data directory structure

The Surface data is organized as xml files every 15 minutes :

```
> ls -la /g3/tianwh/share/conobs/rec_RSURF_20190*
-rw-r--r-- 1 tianwh obs 11482070 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060100_g_06.dat
-rw-r--r-- 1 tianwh obs 11654320 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060106_g_06.dat
-rw-r--r-- 1 tianwh obs 11637320 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060112_g_06.dat
-rw-r--r-- 1 tianwh obs 11630570 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060118_g_06.dat
-rw-r--r-- 1 tianwh obs 11549820 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060200_g_06.dat
-rw-r--r-- 1 tianwh obs 11672570 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060206_g_06.dat
-rw-r--r-- 1 tianwh obs 11687820 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060212_g_06.dat
-rw-r--r-- 1 tianwh obs 11619320 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060218_g_06.dat
-rw-r--r-- 1 tianwh obs 11566570 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060300_g_06.dat
-rw-r--r-- 1 tianwh obs 11562820 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060306_g_06.dat
-rw-r--r-- 1 tianwh obs 11619320 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060312_g_06.dat
-rw-r--r-- 1 tianwh obs 11468820 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060318_g_06.dat
-rw-r--r-- 1 tianwh obs 11398070 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060400_g_06.dat
```

(下页继续)

(续上页)

```
-rw-r--r-- 1 tianwh obs 11605820 9 月  25 00:59 /g3/tianwh/share/conobs/rec_RSURF_
↪2019060406_g_06.dat
......
```

Data format

CIMISS format data is looks like:

```
RSURF    2019       6       1       0   45928       6       9       9
00081        81.00     -5.53    -36.87     19.00      1.00    100.00
   1013.20     23.05     21.35    347.00      0.00 999999.00 999999.00 999999.00   1011.
↪00      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00  ␣
↪ 8.00
00081        81.00     -5.53    -36.87     19.00      1.00      0.00
   1012.80     24.15     21.75    345.00      0.10 999999.00 999999.00 999999.00   1010.
↪60      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00  ␣
↪ 8.00
00081        81.00     -5.53    -36.87     19.00      1.00    200.00
   1012.90     22.25     21.25    333.00      0.00 999999.00 999999.00 999999.00   1010.
↪70      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00  ␣
↪ 8.00
00086        86.00    -19.98    -43.96   1214.00      1.00      0.00
   1016.00     20.75     14.25    289.00      1.80 999999.00 999999.00 999999.00    882.
↪20      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00  ␣
↪ 8.00
00086        86.00    -19.98    -43.96   1214.00      1.00    100.00
   1016.30     20.25     14.35    306.00      1.50 999999.00 999999.00 999999.00    882.
↪30      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00  ␣
↪ 8.00
00086        86.00    -19.98    -43.96   1214.00     31.00   2100.00
   1015.10     21.35     13.95    288.00      3.90 999999.00 999999.00 999999.00    881.
↪70      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00  ␣
↪ 8.00
00086        86.00    -19.98    -43.96   1214.00      1.00    200.00
   1016.30     20.45     14.35    283.00      2.50 999999.00 999999.00 999999.00    882.
↪40      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00  ␣
↪ 8.00
01001      1001.00     70.94     -8.67     10.00     31.00   2200.00
   1015.80     -0.40     -1.80    341.00     10.70 999999.00 999999.00 999999.00   1014.
↪60      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00  ␣
↪ 0.00
```

(下页继续)

(续上页)

```
01001      1001.00     70.94     -8.67     10.00      1.00    200.00
   1014.60     -0.40     -1.97    337.00     10.30 999999.00 999999.00 999999.00   1013.
↪40      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01001      1001.00     70.94     -8.67     10.00      1.00    100.00
   1015.00     -0.40     -2.06    340.00     11.60 999999.00 999999.00 999999.00   1013.
↪80      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01001      1001.00     70.94     -8.67     10.00     31.00   2300.00
   1015.20     -0.40     -2.23    341.00     10.40 999999.00 999999.00 999999.00   1014.
↪00      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01001      1001.00     70.94     -8.67     10.00     31.00   2100.00
   1015.80     -0.20     -1.89    340.00     13.60 999999.00 999999.00 999999.00   1014.
↪60      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01001      1001.00     70.94     -8.67     10.00      1.00      0.00
   1015.00     -0.20     -1.58    344.00     12.10 999999.00      0.00 999999.00   1013.
↪80      0.00      0.00      0.00      0.00      0.00      8.00      0.00      8.00
↪ 0.00
01002      1002.00     80.06     16.25      8.00      1.00    100.00
   1020.50     -1.90     -2.89    109.00      6.80 999999.00 999999.00 999999.00   1019.
↪70      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01002      1002.00     80.06     16.25      8.00     31.00   2200.00
   1020.80     -0.20     -2.30     89.00      6.90 999999.00 999999.00 999999.00   1020.
↪00      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01002      1002.00     80.06     16.25      8.00     31.00   2300.00
   1020.70     -1.20     -2.59    111.00      6.30 999999.00 999999.00 999999.00   1019.
↪80      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01002      1002.00     80.06     16.25      8.00     31.00   2100.00
   1020.70      0.00     -1.96     85.00      7.20 999999.00 999999.00 999999.00   1019.
↪90      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01002      1002.00     80.06     16.25      8.00      1.00      0.00
   1020.70     -1.90     -3.02    119.00      7.30 999999.00 999999.00 999999.00   1019.
↪80      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
```

(下页继续)

(续上页)

```
01002      1002.00     80.06     16.25      8.00      1.00    200.00
   1020.30     -1.70     -2.96    111.00      8.60 999999.00 999999.00 999999.00   1019.
↪50      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01003      1003.00     77.00     15.54     10.00      1.00    100.00
   1014.70     -1.30     -2.83    110.00     13.70 999999.00 999999.00 999999.00   1013.
↪30      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01003      1003.00     77.00     15.54     10.00     31.00   2300.00
   1014.40      1.50     -3.16     62.00      7.60 999999.00 999999.00 999999.00   1013.
↪00      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01003      1003.00     77.00     15.54     10.00     31.00   2200.00
   1014.00      1.40     -3.26     59.00     12.50 999999.00 999999.00 999999.00   1012.
↪60      0.00      0.00      0.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01003      1003.00     77.00     15.54     10.00     31.00   2100.00
   1013.50      1.40     -3.45     57.00     13.10 999999.00 999999.00 999999.00   1012.
↪10      0.00      1.00      1.00      0.00      0.00      8.00      8.00      8.00
↪ 0.00
01003      1003.00     77.00     15.54     10.00      1.00      0.00
   1014.40      1.10     -2.99     53.00      9.80 999999.00      0.00 999999.00   1013.
↪00      0.00      0.00      0.00      0.00      0.00      8.00      0.00      8.00
↪ 0.00
```

Data extractor

A python code is used to extract the desired information from this CIMISS file:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders
> rm cimiss_data_surface
> ./read_nmc_cimiss_rsurf_rgwst.py -f  /g3/tianwh/share/conobs/rec_RSURF_2019060406_g_06.
↪dat
```

If you want to batch process number of CIMISS files, you can use following command:

```
> # This command will find all CIMISS files and prcessing the file one by one
> rm cimiss_data_surface
> decod_nmc_cimiss_rsing.py -d /g3/tianwh/share/conobs
```

The information we want to extract from CIMISS is:

```
import fortranformat as ff
output_line = ff.FortranRecordWriter('(I8, 3f10.2, 5A8, 2f10.2, 2I8, f10.2)')
output.write(output_line.write([id, lat, lon, elv, obstime.strftime("%Y"), obstime.
↪strftime("%m"), \
        obstime.strftime("%d"), obstime.strftime("%H"), obstime.strftime("%M"), \
        p, T, round(rh), round(dd), ff]) + "\n")
```

the content of `cimiss_data_surface` is:

```
> less cimiss_data_surface
  56076 33.85 102.6333 3429 2018 9 1 16 0 674.7 10.8 97 149 3.4
  674002 40.6644 109.6661 1052 2018 9 1 16 0 890.2 15 95 2 1.3
  735904 31.0231 118.4314 9 2018 9 1 16 0 999999 26.4 999999 158 1.7
  868552 34.55 109.2975 357 2018 9 1 16 0 965.8 24.9 74 356 0.9
  718262 22.8133 112.4264 49 2018 9 1 16 0 999999 25.4 999999 225 0
  708128 26.8458 116.7783 343 2018 9 1 16 0 970.6 22.9 96 176 1.1
  737152 30.7728 117.3278 28.3 2018 9 1 16 0 1005.9 26.3 97 135 0.6
  756060 29.1036 119.8056 58 2018 9 1 16 0 999999 26.5 999999 210 1.1
  804130 27.4056 110.4781 200 2018 9 1 16 0 999999 23.1 999999 999999 999999
  672002 43.2972 116.1158 1284 2018 9 1 16 0 860 13.9 94 217 7.6
  663689 40.2278 119.22 300 2018 9 1 16 0 999999 19.2 999999 999999 999999
  752628 30.1492 121.1961 5 2018 9 1 16 0 999999 28.1 999999 237 1.6
  860410 34.9769 108.5269 1193 2018 9 1 16 0 999999 19.8 999999 999999 999999
  868283 34.2217 108.5681 394.5 2018 9 1 16 0 959.8 26.9 64 302 1
  754019 29.9078 120.4844 59 2018 9 1 16 0 999999 24.6 999999 324 0
  878120 36.0133 108.1183 1129 2018 9 1 16 0 876.3 17.5 99 329 1.4
  751675 29.6344 119.5692 95 2018 9 1 16 0 999999 24.7 999999 333 0
```

Decoder source code

1. Source code directory:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimisssurf/sorc
```

2. Subroutines to decode surface data

   - `lsdcod.f`

注解:

- The *cimiss.tbl* and `decod_WMO.Res40.headers` are not used, although they are required as arguments and read in.

- The path and file name of `cimiss_data_surface` file are hard coded in the subroutines.

4. Compile the code

```
> make
```

Decode and convert to BUFR format

1. enter into the exec directory

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimisssurf/exec
> ls -la
total 2028
drwxr-xr-x. 3 zwtd zwtd     211 Dec 20 22:45 .
drwxr-xr-x. 6 zwtd zwtd      82 Dec 20 22:42 ..
lrwxrwxrwx. 1 zwtd zwtd      34 Dec  9 23:48 bufrtab.000 -> ../../decod_shared/fix/
↪bufrtab.000
-rw-r--r--. 1 zwtd zwtd 1121683 Dec  9 23:48 cimiss.tbl
lrwxrwxrwx. 1 zwtd zwtd      31 Dec  9 23:48 decod_WMO.Res40.headers -> ../parm/decod_
↪WMO.Res40.headers
-rwxr-xr-x. 1 zwtd zwtd  940696 Dec 20 00:24 decod_dccimiss
-rw-r--r--. 1 zwtd zwtd     462 Dec 20 22:45 decod_dccimiss.log
-rwxr-xr-x. 1 zwtd zwtd     484 Dec 20 00:30 run.ksh
-rwxr-xr-x. 1 zwtd zwtd     776 Dec 20 22:44 run_dccimisssurf.py
drwxr-xr-x. 2 zwtd zwtd      58 Dec 20 22:45 tmp
```

2. we provide a script to run the decoder in batch mode:

```
> ./run_dccimisssurf.py -s 2018121600 -e 2018121700 -i 1
```

注解:

- given the starting datetime and ending datetime, it iterates all cycles (every 1 hours)
- the units of interval is hour (-i)
- this script call run.ksh

3. run the decoder script

```
> run.ksh
```

(下页继续)

(续上页)

```
> cat run.ksh
!/bin/bash
export DBNBUFRT=120
export TRANJB=/g5/nwpc_ep8/obsproc/nwprod/ush/tranjb
export tank_dir=/g5/nwpc_ep8/obsproc/nwprod/dcom/us007003
export DBNROOT=`pwd`
rm tmp/*
rm decod_dccimiss.log
./decod_dccimiss -d decod_dccimiss.log -b 240 -c $1 bufrtab.000 cimiss.tbl decod_WMO.
↪Res40.headers
ls -la tmp/*

BUFR_FILES=$(echo tmp/BUFR*)
echo ${BUFR_FILES}

for file in ${BUFR_FILES}
do
  ${TRANJB} ${tank_dir} ${file}
done
```

注解:

- -c $1 : Set the **current time** (201809011600) used to calculate the time departures of the obs. data.
- -b 240 : Number of hours to decode prior to "current" time (default)
- The observations with date/time between **current time** - 240 hours and **current time** + 3 are **kept**.

4. The generated BUFR format file will be saved at

```
> ls -la tmp
-rw-r--r--  1 xinzhang  staff  4199744 Sep 21 18:45 tmp/BUFR.0.cimiss.1.6436.1536097072.8
```

Transfer bufr data to BUFR Tanks

- put data in BUFR **tanks**:

```
> /g5/nwpc_ep8/obsproc/nwprod/ush/tranjb /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003 tmp/
↪BUFR.0.cimiss.1.6436.1536097072.8
```

(下页继续)

(续上页)

```
> ls -al /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/20180901/b000/xx01
-rw-r--r-- 1 vagrant vagrant 4235328 Sep 27 04:42 /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20180901/b000/xx001
```

注解:

- **if environmental variable SCREEN=ON :**
  - Define **Run Time** is the system time when the tranjb is running.
  - Only observations with date/time between **Run Time** - 10 days and **Run Time** + 12 hours are kept.
- for retrospective run, set **SCREEN=OFF**
- `/g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/yyyymmdd/bmmm/xxsss` (where mmm is WMO BUFR message type and xxx is local BUFR message subtype)
- 000.001 (in dump group mnemonic adpsfc): Surface synoptic fixed land reports
- BUFR format
- Arranged by UTC day and continuously grow throughout the day, if you run decoders many time, the content of the file will grow
- No QC (other than rudimentary checks inside decoders)
- No duplicate checking
- Interested users can use utility `debufr` to check the content of the bufr file:

  ```
  > /g5/nwpc_ep8/obsproc/nwprod/util/exec/debufr /g5/nwpc_ep8/obsproc/nwprod/dcom/
  ↪us007003/20180901/b000/xx001
  ```

  the output is in `debufr.out`.

3.10.7 Satwind

Data directory structure

The Satwind data is organized as CIMISS files every 6 hours :

```
> ls -la /g3/tianwh/share/conobs/rec_R2CWE_*
-rw-r--r-- 1 tianwh obs 149671052 9 月  25 01:11 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019073106_g_06.dat
```

(下页继续)

(续上页)

```
-rw-r--r-- 1 tianwh obs 138675992 9 月  25 01:11 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019073112_g_06.dat
-rw-r--r-- 1 tianwh obs 136413932 9 月  25 01:11 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019073118_g_06.dat
-rw-r--r-- 1 tianwh obs 123317130 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081000_g_06.dat
-rw-r--r-- 1 tianwh obs 160374728 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081006_g_06.dat
-rw-r--r-- 1 tianwh obs 147384710 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081012_g_06.dat
-rw-r--r-- 1 tianwh obs 135170580 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081018_g_06.dat
-rw-r--r-- 1 tianwh obs 124427428 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081100_g_06.dat
-rw-r--r-- 1 tianwh obs 145641944 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081106_g_06.dat
-rw-r--r-- 1 tianwh obs 142966948 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081112_g_06.dat
-rw-r--r-- 1 tianwh obs 137905074 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081118_g_06.dat
-rw-r--r-- 1 tianwh obs 127579260 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081200_g_06.dat
-rw-r--r-- 1 tianwh obs 147363836 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081206_g_06.dat
-rw-r--r-- 1 tianwh obs 135758886 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081212_g_06.dat
-rw-r--r-- 1 tianwh obs 139330470 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081218_g_06.dat
-rw-r--r-- 1 tianwh obs 127418090 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081300_g_06.dat
-rw-r--r-- 1 tianwh obs 150981286 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081306_g_06.dat
-rw-r--r-- 1 tianwh obs 141173204 8 月  14 06:13 /g3/tianwh/share/conobs/rec_R2CWE_
↪2019081312_g_06.dat
```

Data format

CIMISS format data is looks like:

```
R2CWE    2019       8      13      12  994177       6       4       4
      4.00     86.08    132.61     13.00   1350.00      1.00
    925.00      3.90    183.00     10.10      0.00      0.00     62.00     62.00
      4.00     86.16    133.38     13.00   1350.00      1.00
    925.00      3.80    184.00     10.60      0.00      0.00     55.00     55.00
      4.00     83.45    109.55     13.00   1350.00      1.00
    925.00      4.40    153.00      8.70      0.00      0.00     59.00     59.00
      4.00     87.66    168.14     13.00   1350.00      1.00
    925.00      2.90    225.00     15.70      0.00      0.00     57.00     57.00
      4.00     83.00    109.08     13.00   1350.00      1.00
    925.00      4.40    149.00      7.00      0.00      0.00     88.00     88.00
      4.00     85.73    142.15     13.00   1350.00      1.00
    925.00      3.40    187.00     11.30      0.00      0.00     61.00     61.00
      4.00    -71.85   -151.74     13.00   1259.00      1.00
    925.00    -12.60     26.00      5.90      0.00      0.00     79.00     79.00
      4.00     85.97    137.32     13.00   1350.00      1.00
    925.00      3.70    184.00     12.70      0.00      0.00     65.00     65.00
      4.00     85.86    136.51     13.00   1350.00      1.00
    925.00      3.70    182.00     12.40      0.00      0.00     87.00     87.00
      4.00     83.35    115.16     13.00   1350.00      1.00
    925.00      4.40    162.00     12.20      0.00      0.00     64.00     64.00
      4.00    -77.50    177.80     13.00   1259.00      1.00
    925.00    -23.60    174.00      7.00      0.00      0.00     79.00     79.00
      4.00    -76.09    174.07     13.00   1259.00      1.00
    925.00    -21.70    175.00      6.90      0.00      0.00     78.00     78.00
      4.00    -78.32    173.49     13.00   1259.00      1.00
    925.00    -25.00    195.00     10.00      0.00      0.00     82.00     82.00
      4.00    -77.63    177.07     13.00   1259.00      1.00
    925.00    -24.10    157.00      9.60      0.00      0.00     85.00     85.00
      4.00    -78.71    172.48     13.00   1259.00      1.00
    925.00    -25.40    204.00      8.10      0.00      0.00     86.00     86.00
      4.00     84.73    110.09     13.00   1350.00      1.00
    925.00      4.20    185.00      8.20      0.00      0.00     74.00     74.00
      4.00     86.67    126.54     13.00   1350.00      1.00
    925.00      3.90    190.00     12.10      0.00      0.00     77.00     77.00
      4.00     84.49    107.68     13.00   1350.00      1.00
    925.00      4.10    156.00      7.90      0.00      0.00     89.00     89.00
      4.00     84.79    108.30     13.00   1350.00      1.00
    925.00      4.10    175.00      7.60      0.00      0.00     64.00     64.00
      4.00     71.52     33.12     13.00   1350.00      1.00
    925.00      3.60     72.00      7.30      0.00      0.00     76.00     76.00
```

Data extractor

A python code is used to extract the desired information from this CIMISS file:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders
> rm satob_data
> ./read_nmc_cimiss_r2cwe.py -f /g3/tianwh/share/conobs/rec_R2CWE_2019060100_g_06.dat
```

If you want to batch process number of CIMISS files, you can use following command:

```
> # This command will find all CIMISS files and prcessing the file one by one
> decod_nmc_cimiss_r2cwe.py -d /g3/tianwh/share/conobs
```

The information we want to extract from CIMISS is:

```
import fortranformat as ff
output_line = ff.FortranRecordWriter('(I8, 2f10.2, 6A8, 8f10.2)')
output.write(output_line.write([sat_id, lat, lon, obstime.strftime("%Y"), obstime.
↪strftime("%m"), \
        obstime.strftime("%d"), obstime.strftime("%H"), obstime.strftime("%M"), obstime.
↪strftime("%S"), \
        p, T, dd, ff, flag_p, flag_T, flag_dd, flag_ff]) + "\n")
```

the content of `satob_data` is:

```
> less satob_data
   4    -68.97    176.83    2019      06      12      19      28      00    925.00    -
↪17.90    295.00      8.50      0.00      0.00     70.00     70.00
   4     69.24     32.64    2019      06      12      16      56      00    925.00
↪0.90    309.00      6.00      0.00      0.00     72.00     72.00
   4    -68.75   -177.07    2019      06      12      19      28      00    925.00    -
↪17.70    288.00      9.70      0.00      0.00     86.00     86.00
   4    -69.09    173.94    2019      06      12      19      28      00    925.00    -
↪16.80    314.00      6.20      0.00      0.00     65.00     65.00
   4    -69.00    174.49    2019      06      12      19      28      00    925.00    -
↪16.80    287.00      8.30      0.00      0.00     72.00     72.00
   4    -68.84    173.85    2019      06      12      19      28      00    925.00    -
↪16.10    317.00      7.40      0.00      0.00     63.00     63.00
   4    -68.66    173.45    2019      06      12      19      28      00    925.00    -
↪15.60    317.00      7.30      0.00      0.00     69.00     69.00
   4    -68.78    175.21    2019      06      12      19      28      00    925.00    -
↪16.60    286.00      7.30      0.00      0.00     59.00     59.00
   4    -82.24    177.30    2019      06      12      17      47      00    925.00    -
↪26.80    146.00      5.90      0.00      0.00     81.00     81.00
```

(下页继续)

(续上页)

```
   4    -68.80    178.25    2019      06      12      19      28      00    925.00    -
↪18.20    310.00      6.60      0.00      0.00    67.00    67.00
   4    -67.15     65.80    2019      06      12      17      47      00    925.00    -
↪19.70    108.00     15.20      0.00      0.00    79.00    79.00
   4    -68.41   -175.55    2019      06      12      19      28      00    925.00    -
↪16.80    296.00      9.30      0.00      0.00    99.00    99.00
   4    -69.20   -171.10    2019      06      12      19      28      00    925.00    -
↪16.20    264.00      8.00      0.00      0.00    91.00    91.00
   4    -69.46    173.82    2019      06      12      19      28      00    925.00    -
↪17.80    304.00      7.10      0.00      0.00    75.00    75.00
   4    -66.98     61.89    2019      06      12      17      47      00    925.00    -
↪19.20    110.00     15.80      0.00      0.00    61.00    61.00
   4    -69.36   -173.16    2019      06      12      19      28      00    925.00    -
↪16.10    283.00      7.90      0.00      0.00    65.00    65.00
   4    -69.46   -172.43    2019      06      12      19      28      00    925.00    -
↪16.20    278.00      8.10      0.00      0.00    69.00    69.00
   4    -68.89   -170.92    2019      06      12      19      28      00    925.00    -
↪16.00    262.00      8.60      0.00      0.00    77.00    77.00
   4    -66.85     65.16    2019      06      12      17      47      00    925.00    -
↪19.40     99.00     14.90      0.00      0.00    68.00    68.00
   4     76.08     -3.47    2019      06      12      20      19      00    925.00    ␣
↪1.80    204.00      6.10      0.00      0.00    60.00    60.00
   4    -67.30     71.55    2019      06      12      17      47      00    925.00    -
↪21.80    130.00     10.80      0.00      0.00    77.00    77.00
   4    -69.06    176.09    2019      06      12      19      28      00    925.00    -
↪17.70    285.00      9.20      0.00      0.00    90.00    90.00
   4    -69.48    175.63    2019      06      12      19      28      00    925.00    -
↪18.30    333.00      5.50      0.00      0.00    64.00    64.00
   4    -69.15    175.81    2019      06      12      19      28      00    925.00    -
↪17.70    284.00      9.10      0.00      0.00    95.00    95.00
   4     62.95     46.13    2019      06      12      16      56      00    925.00    ␣
↪0.40    346.00      8.20      0.00      0.00    93.00    93.00
   4     67.73     32.13    2019      06      12      16      56      00    925.00    ␣
↪2.10    319.00      4.50      0.00      0.00    73.00    73.00
   4    -75.74    178.85    2019      06      12      17      47      00    925.00    -
↪22.10    206.00      4.20      0.00      0.00    59.00    59.00
   4    -67.92   -175.35    2019      06      12      19      28      00    925.00    -
↪16.80    289.00      9.50      0.00      0.00    96.00    96.00
   4    -75.81   -163.69    2019      06      12      17      47      00    925.00    -
↪18.10    218.00      8.40      0.00      0.00    70.00    70.00
```

(下页继续)

(续上页)

```
   4      64.16      42.60      2019       06       12       16       56       00      925.00
↪0.70     332.00       7.20       0.00       0.00      68.00      68.00
   4      67.10      31.29      2019       06       12       16       56       00      925.00
↪2.80     318.00       4.00       0.00       0.00      67.00      67.00
```

Decoder source code

1. Source code directory:

```
> cd /g5/nwpc_ep8/obsproc/nwprod/decoders/decod_dccimisssatwind/sorc
```

2. Subroutines to decode Satwind data

   - `eldcod.f`

3. Compile the code:

```
> make
```

Decode and convert to BUFR format

1. enter into the exec directory:

2. The generated BUFR format file will be saved at:

```
> ls -la tmp
-rw-rw-r-- 1 nwpc_ep8 nwpep 71026168 10 月  28 19:23 BUFR.0.elrw.1.107627.1572290433.207
-rw-rw-r-- 1 nwpc_ep8 nwpep 80434768 10 月  28 19:27 BUFR.0.elrw.2.107627.1572290639.361
```

Transfer bufr data to BUFR Tanks

- put data in BUFR **tanks**:

```
> /g5/nwpc_ep8/obsproc/nwprod/ush/tranjb /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003 BUFR.
↪0.elrw.1.107627.1572290433.207

> ls -al /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/20190601/b005/xx064
--rw-r--r-- 1 nwpc_ep8 nwpep 940643952 10 月  28 06:48 /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20190601/b005/xx064
```

注解:

- **if environmental variable SCREEN=ON :**
  - Define **Run Time** is the system time when the tranjb is running.
  - Only observations with date/time between **Run Time** - 10 days and **Run Time** + 12 hours are kept.
- for retrospective run, set **SCREEN=OFF**
- /g5/nwpc_ep8/obsproc/nwprod/dcom/us007003/yyyymmdd/bmmm/xxsss (where mmm is WMO BUFR message type and xxx is local BUFR message subtype)
- BUFR format
- Arranged by UTC day and continuously grow throughout the day, if you run decoders many time, the content of the file will grow
- No QC (other than rudimentary checks inside decoders)
- No duplicate checking
- Interested users can use utility *debufr* to check the content of the bufr file:

```
> /g5/nwpc_ep8/obsproc/nwprod/util/exec/debufr /g5/nwpc_ep8/obsproc/nwprod/dcom/
↪us007003/20190601/b005/xx064
```

the output is in `debufr.out`.

3.11 dump

3.11.1 Dump

Purpose

- BUFR format
- Generated from "TANK" at each network data cutoff time
- Time-windowed
  - NET=gfs : +/- 3.0 hour
  - NET=rap : +/- 1.5 hour or 1 hour
- Geographically filtered
  - environmental variable "**LALO=0**" is used to turn off the filter
- Duplicate checked
- QC: SDM purge (or keep) flags, reject list flags

  - `${ROOT_DIR}/nwprod/dcom/us007003/sdmedit` can be used to filtered out specific reports

- The product is `${ROOT_DIR}/nwprod/com/NET/prod/RUN.yyyymmdd/MODEL.tcycz.TYPE.tmMM.bufr_d`, where:
  - **NET(RUN/MODEL)** is either: cdas(cdas/cdas), cfs(cdas/cdas1),dump(dump/dump), gfs(gdas/gdas1), gfs(gfs/gfs), nam(nam/nam), nam(ndas/ndas), rap(rap/rap), rap(rap_p/rap_p), rtma(rtma/rtma)
  - **cyc** is cycle (hourly for NET=dump, rap, rtma; 00, 06, 12, 18 all others)
  - **MM** is 00 for all types except RUN/MODEL=ndas/ndas where it can be 12, 09, 06 or 03
  - **TYPE** is either: 1bamua, 1bamub, 1bhrs3, 1bhrs4, 1bmhs, adpsfc, adpupa, aircar, aircft, airsev, ascatt, ascatw, atms, atovs, avcsam, avcspm, bathy, cris, esamua, esamub, eshrs3, esmhs, geoimr, goesfv, goesnd, gome, gpsipw, gpsro, lghtng, lgycld, mls, msonet, mtiasi, nexrad, omi, osbuv8, proflr, radwnd, rassda, satwnd, sevcsr, sfcshp, sptrmm, ssmisu, tesac, trkob, trmm, vadwnd, wdsatr, wndsat (types actually dumped depend upon the network)

3.11.2 运行步骤

1. 在 `${ROOT_DIR/workflow/include/head.h` 中定义如下变量，通知系统 **dump** 哪些观测

```
# JOB_NUMBER = 1 indicates the prepbufr dump job.
# JOB_NUMBER = 2 indicates the non-prepbufr dump job.
# JOB_NUMBER not present indicates dump BOTH prepbufr and non-prepbufr data.
# --------------------------------------------------------------------------
# Dump group #1 (non-pb) = 1bamua 1bmhs radwnd esamua esmhs atms mtiasi␣
↪sevcsr
#                          gpsro esiasi iasidb esatms atmsdb sevasr amsr2
# Dump group #2 (pb) = vadwnd satwnd adpupa
# Dump group #3 (pb) = proflr rassda sfcshp adpsfc ascatt
# Dump group #4 (pb) = msonet gpsipw
# Dump group #5 (pb) = aircft aircar goesnd
# Dump group #6 (non-pb) = nexrad
# Dump group #7 (non-pb) = 1bhrs4 airsev lghtng eshrs3 lgycld cris ssmisu␣
↪osbuv8
#                          escris crisdb saphir
# Dump group #8 STATUS FILE
```

2. `${ROOT_DIR}/nwprod/obsproc_rap.v${obsproc_rap_ver}/scripts/exrap_dump.sh.ecf`

   根据 `Dump group` 的设置，产生若干个临时脚本（如：`thread_2`, `thread_3`，分别对应 `Dump group #2` 和 `Dump group #3`）

3. `thread_x` 会分别提交到后台运行，分别调用 ${ROOT_DIR}/nwprod/obsproc_dump.${obsproc_dump_ver}/ush/buf

- thread_2 会调用：bufr_dump_obs.sh 20012922 2.5 1 vadwnd satwnd adpupa

  再调用 dumpjb 20012922 2.5 vadwnd satwnd adpupa

  注解： 对每种资料的时间窗口还会根据资料的类型调整

  从 ${ROOT_DIR}/dcom/us007003/ 中读取 *2020-01-29_22Z +/-1.0 hour* 的上述 3 种观测资料，例如：对 *adpupa*：

```
Dumping *002.001* in data group adpupa for 2020012921.00-2020012923.00
Domain for 002.001 in data group adpupa : 1- N lat, - W lon
 ===> Background Processing is set to ON

Missing *002.002* in data group adpupa for 2020012921.00-2020012923.00
Domain for 002.002 in data group adpupa : 1- N lat, - W lon

Missing *002.003* in data group adpupa for 2020012921.00-2020012923.00
Domain for 002.003 in data group adpupa : 1- N lat, - W lon

Missing *002.004* in data group adpupa for 2020012921.00-2020012923.00
Domain for 002.004 in data group adpupa : 1- N lat, - W lon

Dumping *002.005* in data group adpupa for 2020012921.00-2020012923.00
Domain for 002.005 in data group adpupa : 1- N lat, - W lon
 ===> Background Processing is set to ON

Dumping *004.005* in data group adpupa for 2020012921.00-2020012923.00
Domain for 004.005 in data group adpupa : 1- N lat, - W lon
 ===> Background Processing is set to ON
```

- thread_3 会调用：bufr_dump_obs.sh 20012922 0.5 1 proflr rassda sfcshp adpsfc ascatt

  再调用 dumpjb 20012922 0.5 proflr rassda sfcshp adpsfc ascatt

  从 ${ROOT_DIR}/dcom/us007003/ 中读取 *2020-01-29_22Z +/-0.5 hour* 的上述 5 种观测资料，例如：对 *sfcshp*：

```
Missing *001.001* in data group sfcshp for 2020012921.50-2020012922.50
Domain for 001.001 in data group sfcshp : 1- N lat, - W lon

Dumping *001.013* in data group sfcshp for 2020012921.50-2020012922.50
Domain for 001.013 in data group sfcshp : 1- N lat, - W lon
```

（下页继续）

(续上页)

```
 ===> Background Processing is set to ON

Dumping *001.002* in data group sfcshp for 2020012921.50-2020012922.50
Domain for 001.002 in data group sfcshp : 1- N lat, - W lon
 ===> Background Processing is set to ON

Dumping *001.003* in data group sfcshp for 2020012921.50-2020012922.50
Domain for 001.003 in data group sfcshp : 1- N lat, - W lon
 ===> Background Processing is set to ON

Missing *001.004* in data group sfcshp for 2020012921.50-2020012922.50
Domain for 001.004 in data group sfcshp : 1- N lat, - W lon

Missing *001.005* in data group sfcshp for 2020012921.50-2020012922.50
Domain for 001.005 in data group sfcshp : 1- N lat, - W lon

Missing *001.007* in data group sfcshp for 2020012921.50-2020012922.50
Domain for 001.007 in data group sfcshp : 1- N lat, - W lon
```

对 *sfcshp*：

```
Dumping *000.000* in data group adpsfc for 2020012921.50-2020012922.50
Domain for 000.000 in data group adpsfc : 1- N lat, - W lon
 ===> Background Processing is set to ON

Dumping *000.001* in data group adpsfc for 2020012921.50-2020012922.50
Domain for 000.001 in data group adpsfc : 1- N lat, - W lon
 ===> Background Processing is set to ON
Skipped 000.002 in data group adpsfc as directed

Dumping *000.007* in data group adpsfc for 2020012921.50-2020012922.50
Domain for 000.007 in data group adpsfc : 1- N lat, - W lon
 ===> Background Processing is set to ON
```

- thread_5 会调用：`bufr_dump_obs.sh 20012922 3.25 1 aircft aircar goesnd`

  再调用 `dumpjb 20012922 3.25 aircft aircar goesnd`

  从 `${ROOT_DIR}/dcom/us007003/` 中读取 *2020-01-29_22Z +/-3.25 hour* 的上述 5 种观测资料，例如：对 *aircft*：

```
Dumping *004.001* in data group aircft for 2020012918.75-2020013001.25
Domain for 004.001 in data group aircft : Global (NO lat/lon filtering)
 ===> Background Processing is set to ON

Dumping *004.002* in data group aircft for 2020012918.75-2020013001.25
Domain for 004.002 in data group aircft : Global (NO lat/lon filtering)
 ===> Background Processing is set to ON

Dumping *004.003* in data group aircft for 2020012918.75-2020013001.25
Domain for 004.003 in data group aircft : Global (NO lat/lon filtering)
 ===> Background Processing is set to ON

Missing *004.006* in data group aircft for 2020012918.75-2020013001.25
Domain for 004.006 in data group aircft : Global (NO lat/lon filtering)

Missing *004.009* in data group aircft for 2020012918.75-2020013001.25
Domain for 004.009 in data group aircft : Global (NO lat/lon filtering)

Missing *004.010* in data group aircft for 2020012918.75-2020013001.25
Domain for 004.010 in data group aircft : Global (NO lat/lon filtering)

Missing *004.011* in data group aircft for 2020012918.75-2020013001.25
Domain for 004.011 in data group aircft : Global (NO lat/lon filtering)

Missing *004.103* in data group aircft for 2020012918.75-2020013001.25
Domain for 004.103 in data group aircft : Global (NO lat/lon filtering)
```

4. 对每种观测资料的每个子类文件运行 BUFR_DUMPMD, 产生如下输出:

```
 ---> Welcome to BUFR_DUMPMD - Version 01-13-2013

REQUESTED EARLIEST DATE IS ...  2020012921.00000000
REQUESTED LATEST   DATE IS ...  2020012923.00000000
REQUESTED CENTER   DATE IS ...  2020012922.00000000

DATABASE INPUT FILE 1 IS
     /sya/u/hailang/chinawise/dcom/us007003/20200129/b002/xx001
DATABASE INPUT FILE 2 IS
     NONE
TIME WINDOWED OUTPUT DUMP FILE IS
     /tmp/dumpjb.sya05n13.20200130025545.5505184/adpupa/dird/01_002.001
```

(下页继续)

(续上页)

```
++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++
BUFRLIB: OPENBF - DEGREE OF MESSAGE PRINT INDICATOR CHNGED FROM  0 (limited␣
↪- default)         TO  1 (all warnings)
++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++


NPUT BUFR FILE 1 MESSAGES   U N C O M P R E S S E D
IRST MESSAGE TYPE FOUND IS    2

++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++
BUFRLIB: OPENBF - DEGREE OF MESSAGE PRINT INDICATOR CHNGED FROM  1 (all␣
↪warnings)              TO  0 (limited - default)
++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++


   181 MESSAGES READ IN  FROM DATABASE

    25 MESSAGES WINDOWED FROM DATABASE


UTPUT BUFR FILE MESSAGES   U N C O M P R E S S E D
IRST MESSAGE TYPE FOUND IS    2
```

5. 对每种观测资料的每个子类文件运行 BUFR_GEOFIL, 产生如下输出:

```
UNFILTERED INPUT FILE IS
     /tmp/dumpjb.sya05n13.20200130025545.5505184/adpupa/dird/01_002.001
WORKSPACE (SCRATCH) FILE IS
     /tmp/dumpjb.sya05n13.20200130025545.5505184/adpupa/dest/01_002.001
GEOGRAPHICALLY FILTERED OUTPUT FILE IS
     /tmp/dumpjb.sya05n13.20200130025545.5505184/adpupa/dird/01_002.001
GEOGRAPHICAL FILTERING PARAMETER FILE IS
     1

GEOGRAPHICAL FILTERING IS PERFORMED HERE USING LAT/LON BOX WITH THE␣
↪FOLLOWING BOUNDARY:
.. NORTHERN LATITUDE (DEG. N+, S-) ......      .0
.. SOUTHERN LATITUDE (DEG. N+, S-) ......    1.0
.. EASTERN LONGITUDE (0-360 W) ..........      .0
```

(下页继续)

(续上页)

```
.. WESTERN LONGITUDE (0-360 W) ..........     .0


INPUT BUFR FILE MESSAGES   U N C O M P R E S S E D
FIRST MESSAGE TYPE FOUND IS    2

 ++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++
 BUFRLIB: OPENBF - DEGREE OF MESSAGE PRINT INDICATOR CHNGED FROM  0␣
↪(limited - default)         TO  1 (all warnings)
 ++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++

 ++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++
 BUFRLIB: READMG - INTERNAL DICTIONARY MESSAGE READ; ACCOUNT FOR IT THEN␣
↪READ IN NEXT MESSAGE WITHOUT RETURNING
 ++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++

 ++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++
 BUFRLIB: OPENBF - DEGREE OF MESSAGE PRINT INDICATOR CHNGED FROM  1 (all␣
↪warnings)              TO  0 (limited - default)
 ++++++++++++++BUFR ARCHIVE LIBRARY+++++++++++++++++


OUTPUT BUFR FILE HAS NO DATA MESSAGES
FIRST MESSAGE TYPE FOUND IS   -2


BUFR_GEOFIL READ IN A TOTAL OF     246 REPORTS

NUMBER OF PASSED REPORTS WRITTEN OUT ...........      0
NUMBER OF REPORTS SKIPPED DUE TO:
   GEOGRAPHIC FILTERING ........................    246
```

3.12 绘图

3.12.1 RAP 的绘图

RAP 的绘图由如下脚本控制:

```
${ROOT_DIR}/nwprod/rap.v4.0.12/jobs/JRAP_NCLPART1
${ROOT_DIR}/nwprod/rap.v4.0.12/jobs/JRAP_NCLPART2

调用:

${ROOT_DIR}/nwprod/rap.v4.0.12/scripts/exrap_nclpart1.sh.ecf
${ROOT_DIR}/nwprod/rap.v4.0.12/scripts/exrap_nclpart2.sh.ecf

这两个脚本调用具体的 *ncl* 脚本
```

具体的 *ncl* 脚本在:

```
cd ${ROOT_DIR}/nwprod/rap.v4.0.12/ncl/ncl/nclrr_rll

[~/nwprod/rap.v4.0.12/ncl/ncl/nclrr_rll]$ls *.ncl
WRFUserARW.ncl          rr_htxs_grmr.ncl     rr_sfc_ltng.ncl     rr_sfc_
↪temp.ncl    yyrr_2m_temp.ncl    yyrr_sfc_pchg.ncl
griblook.ncl            rr_htxs_rwmr.ncl     rr_sfc_mcc.ncl      rr_sfc_
↪totp.ncl    yyrr_500_temp.ncl   yyrr_sfc_ptyp.ncl
plot_hodo.ncl           rr_htxs_sh.ncl       rr_sfc_mnvv.ncl     rr_sfc_
↪ulwrf.ncl   yyrr_700_temp.ncl   yyrr_sfc_pwtr.ncl
rr_10m_gust.ncl         rr_htxs_snmr.ncl     rr_sfc_mucp.ncl     rr_sfc_
↪vis.ncl     yyrr_80m_wind.ncl   yyrr_sfc_rhpw.ncl
rr_10m_wind.ncl         rr_htxs_temp.ncl     rr_sfc_mulcp.ncl    rr_sfc_
↪weasd.ncl   yyrr_850_temp.ncl   yyrr_sfc_shtfl.ncl
rr_10m_wind_track.ncl   rr_htxs_wind.ncl     rr_sfc_mxcp.ncl     rr_ua_
↪ceil.ncl     yyrr_925_temp.ncl   yyrr_sfc_snod.ncl
rr_20cm_soilt.ncl       rr_nta_ulwrf.ncl     rr_sfc_pchg.ncl     rr_ua_
↪ctop.ncl     yyrr_sfc_1hsnw.ncl  yyrr_sfc_solar.ncl
rr_20cm_soilw.ncl       rr_prxs_temp.ncl     rr_sfc_profile.ncl  rr_ua_
↪rh.ncl       yyrr_sfc_acp.ncl    yyrr_sfc_tcc.ncl
rr_2ds_temp.ncl         rr_sfc_1hsnw.ncl     rr_sfc_ptyp.ncl     rr_ua_
↪rh8.ncl      yyrr_sfc_acpcp.ncl  yyrr_sfc_temp.ncl
rr_2m_dewp.ncl          rr_sfc_acp.ncl       rr_sfc_pwtr.ncl     rr_ua_
↪vort.ncl     yyrr_sfc_acsnw.ncl  yyrr_sfc_totp.ncl
rr_2m_ptemp.ncl         rr_sfc_acp_3h.ncl    rr_sfc_rhpw.ncl     rr_ua_
↪vvel.ncl     yyrr_sfc_cape.ncl   yyrr_sfc_vis.ncl
rr_2m_rh.ncl            rr_sfc_acpcp.ncl     rr_sfc_shtfl.ncl    rr_ua_
↪wind.ncl     yyrr_sfc_cin.ncl    yyrr_sfc_weasd.ncl
rr_2m_temp.ncl          rr_sfc_acpcp_3h.ncl  rr_sfc_skewt.ncl    rr_ua_
↪wmag.ncl     yyrr_sfc_cref.ncl   yyrr_ua_ceil.ncl
```

(下页继续)

(续上页)

```
rr_500_temp.ncl        rr_sfc_acsnw.ncl     rr_sfc_skewt1.ncl   skewt_
↪func.ncl     yyrr_sfc_flru.ncl   yyrr_ua_ctop.ncl
rr_5cm_soilt.ncl       rr_sfc_cape.ncl      rr_sfc_skewt2.ncl  ␣
↪soilmois.ncl       yyrr_sfc_hcc.ncl    yyrr_ua_rh.ncl
rr_5cm_soilw.ncl       rr_sfc_cin.ncl       rr_sfc_skewt3.ncl  ␣
↪wrfxsecs.ncl       yyrr_sfc_hpbl.ncl   yyrr_ua_rh8.ncl
rr_700_temp.ncl        rr_sfc_cref.ncl      rr_sfc_skewt4.ncl   yyrr_
↪10m_gust.ncl  yyrr_sfc_lcc.ncl    yyrr_ua_vort.ncl
rr_80m_wind.ncl        rr_sfc_flru.ncl      rr_sfc_snod.ncl     yyrr_
↪10m_wind.ncl  yyrr_sfc_ltng.ncl   yyrr_ua_vvel.ncl
rr_850_temp.ncl        rr_sfc_hcc.ncl       rr_sfc_soilt.ncl    yyrr_
↪2ds_temp.ncl  yyrr_sfc_mcc.ncl    yyrr_ua_wind.ncl
rr_925_temp.ncl        rr_sfc_hpbl.ncl      rr_sfc_soilw.ncl    yyrr_
↪2m_dewp.ncl   yyrr_sfc_mucp.ncl   yyrr_ua_wmag.ncl
rr_bsfc_soilt.ncl      rr_sfc_lcc.ncl       rr_sfc_solar.ncl    yyrr_
↪2m_ptemp.ncl  yyrr_sfc_mulcp.ncl
rr_bsfc_soilw.ncl      rr_sfc_lhtfl.ncl     rr_sfc_tcc.ncl      yyrr_
↪2m_rh.ncl     yyrr_sfc_mxcp.ncl
```

每个脚本负责画出一个图形。绘制的图形统一存放在:

```
${ROOT_DIR}/com/rap/prod/rap.ccyymmdd/nclprd

[/com/rap/prod/rap.20200209/nclprd]$ls -la
total 448
drwxr-xr-x 4 hailang work    512 Feb  9 20:54 .
drwxrwxr-x 4 hailang work 131072 Feb 10 00:12 ..
drwxr-xr-x 2 hailang work  32768 Feb 10 00:02 dfi
drwxr-xr-x 2 hailang work  65536 Feb 10 00:14 full
[hailang@sya01n01 ~/chinawise/com/rap/prod/rap.20200209/nclprd]$ls dfi
cape_sfc.t17z.f00.png  mucp_sfc.t17z.f00.png   pwtr_sfc.t17z.f00.png ␣
↪temp_sfc.t17z.f00.png   wind_850.t17z.f00.png
cape_sfc.t23z.f00.png  mucp_sfc.t23z.f00.png   pwtr_sfc.t23z.f00.png ␣
↪temp_sfc.t23z.f00.png   wind_850.t23z.f00.png
cin_sfc.t17z.f00.png   mulcp_sfc.t17z.f00.png  rh_2m.t17z.f00.png    ␣
↪weasd_sfc.t17z.f00.png  wmag_250.t17z.f00.png
cin_sfc.t23z.f00.png   mulcp_sfc.t23z.f00.png  rh_2m.t23z.f00.png    ␣
↪weasd_sfc.t23z.f00.png  wmag_250.t23z.f00.png
cref_sfc.t17z.f00.png  mxcp_sfc.t17z.f00.png   snod_sfc.t17z.f00.png ␣
↪wind_10m.t17z.f00.png   wmag_850.t17z.f00.png
```

(下页继续)

(续上页)

```
cref_sfc.t23z.f00.png  mxcp_sfc.t23z.f00.png   snod_sfc.t23z.f00.png
↪wind_10m.t23z.f00.png   wmag_850.t23z.f00.png
dewp_2m.t17z.f00.png   ptemp_2m.t17z.f00.png   temp_2ds.t17z.f00.png
↪wind_250.t17z.f00.png
dewp_2m.t23z.f00.png   ptemp_2m.t23z.f00.png   temp_2ds.t23z.f00.png
↪wind_250.t23z.f00.png
gust_10m.t17z.f00.png  ptyp_sfc.t17z.f00.png   temp_2m.t17z.f00.png
↪wind_80m.t17z.f00.png
gust_10m.t23z.f00.png  ptyp_sfc.t23z.f00.png   temp_2m.t23z.f00.png
↪wind_80m.t23z.f00.png
[hailang@sya01n01 ~/chinawise/com/rap/prod/rap.20200209/nclprd]$ls full
acpcp_sfc.t17z.f02.png  hcc_sfc.t17z.f03.png    ptemp_2m.t23z.f01.png
↪ temp_2ds.t23z.f00.png  vis_sfc.t23z.f03.png
acpcp_sfc.t17z.f03.png  hcc_sfc.t17z.f04.png    ptemp_2m.t23z.f02.png
↪ temp_2ds.t23z.f01.png  vis_sfc.t23z.f04.png
acpcp_sfc.t17z.f04.png  hcc_sfc.t17z.f05.png    ptemp_2m.t23z.f03.png
↪ temp_2ds.t23z.f02.png  vis_sfc.t23z.f05.png
acpcp_sfc.t17z.f05.png  hcc_sfc.t17z.f06.png    ptemp_2m.t23z.f04.png
↪ temp_2ds.t23z.f03.png  vis_sfc.t23z.f06.png
acpcp_sfc.t17z.f06.png  hcc_sfc.t23z.f00.png    ptemp_2m.t23z.f05.png
↪ temp_2ds.t23z.f04.png  vvel_700.t17z.f00.png
acpcp_sfc.t23z.f02.png  hcc_sfc.t23z.f01.png    ptemp_2m.t23z.f06.png
↪ temp_2ds.t23z.f05.png  vvel_700.t17z.f01.png
acpcp_sfc.t23z.f03.png  hcc_sfc.t23z.f02.png    pwtr_sfc.t17z.f00.png
↪ temp_2ds.t23z.f06.png  vvel_700.t17z.f02.png
acpcp_sfc.t23z.f04.png  hcc_sfc.t23z.f03.png    pwtr_sfc.t17z.f01.png
↪ temp_2m.t17z.f00.png   vvel_700.t17z.f03.png
acpcp_sfc.t23z.f05.png  hcc_sfc.t23z.f04.png    pwtr_sfc.t17z.f02.png
↪ temp_2m.t17z.f01.png   vvel_700.t17z.f04.png
acpcp_sfc.t23z.f06.png  hcc_sfc.t23z.f05.png    pwtr_sfc.t17z.f03.png
↪ temp_2m.t17z.f02.png   vvel_700.t17z.f05.png
cape_sfc.t17z.f00.png   hcc_sfc.t23z.f06.png    pwtr_sfc.t17z.f04.png
↪ temp_2m.t17z.f03.png   vvel_700.t17z.f06.png
cape_sfc.t17z.f01.png   lcc_sfc.t17z.f00.png    pwtr_sfc.t17z.f05.png
↪ temp_2m.t17z.f04.png   vvel_700.t23z.f00.png
cape_sfc.t17z.f02.png   lcc_sfc.t17z.f01.png    pwtr_sfc.t17z.f06.png
↪ temp_2m.t17z.f05.png   vvel_700.t23z.f01.png
cape_sfc.t17z.f03.png   lcc_sfc.t17z.f02.png    pwtr_sfc.t23z.f00.png
↪ temp_2m.t17z.f06.png   vvel_700.t23z.f02.png
cape_sfc.t17z.f04.png   lcc_sfc.t17z.f03.png    pwtr_sfc.t23z.f01.png
↪ temp_2m.t23z.f00.png   vvel_700.t23z.f03.png
```

(下页继续)

(续上页)

注解: 所有的脚本使用 `${ROOT_DIR}/nwprod/rap.v4.0.12/ncl/ncl/nclrr_rll/names.txt` 中的变量名字典列表来读取所需的变量。可以利用 *ncl* 工具 *ncl_filedump*，来列出 *grib* 文件中所包含的列表名，例如:

```
$ncl_filedump /sya/u/hailang/chinawise/com/rap/prod/rap.20200209/rap.t17z.
↪wrfprsf06.grib2
 Copyright (C) 1995-2013 - All Rights Reserved
 University Corporation for Atmospheric Research
 NCAR Command Language Version 6.1.2
 The use of this software is governed by a License Agreement.
 See http://www.ncl.ucar.edu/ for more details.

Variable: f
Type: file
filename:       rap.t17z.wrfprsf06
path:   /sya/u/hailang/chinawise/com/rap/prod/rap.20200209/rap.t17z.wrfprsf06.
↪grib2
   file global attributes:
   dimensions:
      ygrid_0 = 462
      xgrid_0 = 525
      lv_ISBL0 = 40
      lv_HTGL1 = 2
      lv_SPDL2 = 6
      lv_HTGL3 = 2
      lv_HTGL4 = 2
      lv_TMPL5 = 2
      lv_SPDL6 = 3
      lv_HTGL7 = 2
      lv_HTGL8 = 2
      lv_DBLL9 = 9
   variables:
      float TMP_P0_L1_GLC0 ( ygrid_0, xgrid_0 )
         center :       US National Weather Service - NCEP (WMC)
         production_status :    Operational products
         long_name :    Temperature
         units :        K
```

(下页继续)

(续上页)

```
        _FillValue :    1e+20
        coordinates :   gridlat_0 gridlon_0
        grid_type :     Lambert Conformal can be secant or tangent, conical or␣
↪bipolar
        parameter_discipline_and_category :     Meteorological products,␣
↪Temperature
        parameter_template_discipline_category_number :         ( 0, 0, 0, 0 )
        level_type :    Ground or water surface
        level :          0
        forecast_time :         6
        forecast_time_units :   hours
        initial_time : 02/09/2020 (17:00)

        float TMP_P0_L105_GLC0 ( ygrid_0, xgrid_0 )
           center :        US National Weather Service - NCEP (WMC)
           production_status :     Operational products
           long_name :     Temperature
           units :         K
           _FillValue :    1e+20
           coordinates :   gridlat_0 gridlon_0
           grid_type :     Lambert Conformal can be secant or tangent, conical␣
↪or bipolar
           parameter_discipline_and_category :     Meteorological products,␣
↪Temperature
           parameter_template_discipline_category_number :         ( 0, 0, 0,␣
↪0 )
           level_type :    Hybrid level
           level :          1
           forecast_time :         6
           forecast_time_units :   hours
           initial_time : 02/09/2020 (17:00)

        float TMP_P0_2L108_GLC0 ( lv_SPDL2, ygrid_0, xgrid_0 )
           center :        US National Weather Service - NCEP (WMC)
           production_status :     Operational products
           long_name :     Temperature
           units :         K
           _FillValue :    1e+20
           coordinates :   gridlat_0 gridlon_0
           grid_type :     Lambert Conformal can be secant or tangent, conical␣
↪or bipolar
```

(下页继续)

(续上页)

```
            parameter_discipline_and_category :    Meteorological products,␣
↪Temperature
            parameter_template_discipline_category_number :        ( 0, 0, 0,␣
↪0 )
            level_type :   Level at specified pressure difference from ground␣
↪to level (Pa)
            forecast_time :        6
            forecast_time_units :  hours
            initial_time : 02/09/2020 (17:00)
```

3.12.2 HRRR 的绘图

3.13 如何改变模式模拟区域

用户可能会根据业务需求，或根据计算资源调整预报区域，请按照以下的步骤进行修改：

1. 生成新的地形文件，分别拷入：

```
$ mv geo_em.d01.nc.d1 nwprod/rap.v4.0.12/fix/rap_geo_em.d01.nc

$ mv geo_em.d01.nc.d2 nwprod/hrrr.v3.0.8/fix/conus/hrrr_geo_em.d01.nc
$ cp nwprod/hrrr.v3.0.8/fix/conus/hrrr_geo_em.d01.nc nwprod/hrrr.v3.0.8/ncl/
↪ncl/geo_em.d01.nc
```

2. 修改 namelist

```
$ edit nwprod/rap.v4.0.12/parm/rap_namelist.wps
$ edit nwprod/rap.v4.0.12/parm/rap_wrf.nl
$ edit nwprod/rap.v4.0.12/parm/rap_wrfbc.nl

$ edit nwprod/hrrr.v3.0.8/parm/conus/hrrr_namelist.wps
$ edit nwprod/hrrr.v3.0.8/parm/conus/hrrr_real.nl
$ edit nwprod/hrrr.v3.0.8/parm/conus/hrrr_wrf.nl
$ edit nwprod/hrrr.v3.0.8/parm/conus/hrrr_wrfpre.nl
$ edit nwprod/hrrr.v3.0.8/parm/conus/hrrr_wrfpre_norad.nl
```

3. 修改如下脚本，寻找 *targetsize*, 替换修改后的 *wrfout* 的 size：

```
$ exrap_analysis.sh.ecf:    targetsize=xxxxxxx
$ exrap_analysis_pcyc.sh.ecf:      targetsize=xxxxxxxx
$ exrap_post.sh.ecf:  targetsize=xxxxxx

$ exhrrr_analysis.sh.ecf:targetsize=xxxxxx
$ exhrrr_post.sh.ecf:targetsize=xxxxxx
$ exhrrr_prep_ref2tten.sh.ecf:targetsize=xxxxxx
```

4. 修改 exrap_post.sh.ecf, 参见：wgrib2 文档.

```
$ edit nwprod/rap.v4.0.12/scripts/exrap_post.sh.ecf
...
Modify:
export grid_specs_130="lambert:105:30.000000:60.000000 76.251953:525:13000.
↪000000 13.022026:462:13000.000000"
```

5. 重新生成 enkf 文件

6. 删除原来的初始条件和边界条件

```
$ rm nwges/prod/rap/rapbc/*
$ rm nwges/prod/rap/rapges/*
$ rm nwges/prod/rap/rapges_pcyc/*

$ rm /nwges/prod/hrrr/hrrrges_sfc/conus/*
```

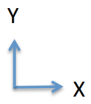
3.14 WRF IO Quilting

- Two types of MPI tasks: compute (client) and I/O (server)
- Compute tasks:
  - Total number = nproc_x*nproc_y (number of processors along X and y axes for decomposion)
  - First number is zero
- I/O tasks:
  - Total number = nio_groups*nio_tasks_per_group
  - Nio_tasks_per_group cannot exceed nproc_y
  - First I/O task number immediately follows last compute task number
- Code will attempt to match each I/O task with compute tasks in east-west rows
  - Ideally nproc_y should be exact multiple of nio_tasks_per_group

Sample Task Layouts

Example compute task layout
nproc_x = 4, nproc_y = 3

8	9	10	11
4	5	6	7
0	1	2	3



If nio_groups=2 and nio_groups_per_task=3, then:

14	8	9	10	11	17
13	4	5	6	7	16
12	0	1	2	3	15

I/O Group 1 I/O Group 2

- Each compute node in a row (e.g., tasks 0-3) will send data to a I/O server (e.g., 12 or 15) at output time.
- One I/O group is selected on-the-fly at output time to handle output to a particular file.
- Within an I/O group, the servers will forward data to designated “root” server which will perform the actual write.

Bibliography

[chenbaode2013] 陈葆德, 王晓峰, 李泓, 张蕾 (2013): 快速更新同化预报的关键技术综述, 气象科技进展 3(2), 29-35