2020-01-23: IODA Data Format

Tom opened the meeting by welcoming the attendees and by noting that, though this is nominally a meeting of the JCSDA Executive Team (ET), we will also be joined by participants from the broader JEDI community who have been invited to participate through the weekly meeting email list.

Yannick then proceeded by outlining the purpose of the meeting and by presenting the following slides, which summarize the current status of the IODA subsystem and the next steps planned for its development.   IODA is the component of JEDI that handles observational data and in particular, how this data is accessed by data assimilation (DA) applications. In addition to an update on the status of IODA for the benefit of the ET, the purpose of this meeting is to discuss what format should be adopted for IODA observational data files as we move forward. We would like to make a decision on this soon and, with this meeting, we are seeking input from the ET and the community of JEDI users and developers.

Please see the slides for details; what follows is a brief summary. Yannick began by emphasizing the **Separation of Concerns** design principle adopted by JEDI. In the present context, this means that whatever decision we make on the format used for IODA data files should be transparent to the general JEDI user.   We have designed IODA to isolate and separate the scientific aspects of applications from the technical details of how the data is stored.   Yannick described three levels of data storage, namely long-term storage, files on disk for a single DA cycle, and in-memory handling of the observational data. A conceptual outline of the latter (in-memory IODA data store) is presented on slide 5. As demonstrated in slide 6, the API for accessing this in-memory data store is the same regardless of what format is used for the data files. In short, **IODA will be a success if users don't need to know what the file format is** (slide 8).

Yannick described the observational data flow used in many current systems as overly complicated, with multiple reads/writes of files and multiple conversions between different data formats (slide 3). He contrasted this with the vision for JEDI, which centers around the IODA data format that will be read directly for preprocessing, DA, and diagnostics. Yannick then stressed again the need to keep the technical implementation separate from the applications in order to allow us to optimally exploit changing memory hardware and software hierarchies as HPC systems continue to evolve (slide 7).

Yannick concluded by introducing the two file formats that we have considered thus far as the leading options for IODA, namely netcdf4 and ODB.   He then announced that Steve H would present these two options in greater detail.

Before Steve H's presentation, we opened the floor for questions. Ricardo brought up Yannick's comparison of a typical observation data flow and the JEDI vision on slides 3 and 4. He said that, in practice, all applications, diagnostics, and pre-processing will likely not access the same IODA data files. Rather, we will likely need to partition the data for some applications such as cycling DA runs. Yannick agreed that this would likely be needed to maximize efficiency.

Ricardo also asked about the internal IODA data representation on slide 5 and in particular whether the IODA variable names are compliant with CF standards. Yannick responded that the CF standards are incomplete with respect to observational data so we need to either extend it or find a different standard. We have not yet made a decision on that.

IodaStatusTestR...ults_012320.pdf

Steve H's presentation focused on results from benchmark testing that he and Steve V have carried out over the past several months on various systems, with various applications and configurations (again, see slides for details). He began by reviewing the IODA requirements that were compiled about a year ago as an outcome of a focused workshop in Monterey, CA. He then described relevant JCSDA GItHub repositories, including ioda itself and a separate repository called ioda-converters that houses many conversion scripts to the ioda data formats now being considered. The implementation details of the file formats are kept isolated from the JEDI applications through the IodaIO Class, which includes NetcdfIO and OdcIO subclasses (ODC is an interface for ODB recently developed by ECMWF).

Steve H then described a benchmark H(x) application that he and Steve V have been using for performance benchmarking and demonstrated that the quantitative results were the same for both netcdf4 and ODB/ODC. He then focused on a comparison of the efficiency. The overall performance of the two formats was similar, though Steve H highlighted a few differences.   In particular, the execution time to read large files (>~ 100 MB) was significantly longer for netcdf4 (slide 9). Will remarked that 157 MB (the largest size considered in the tests) is not that large and many of the obs files in the future are likely to be bigger.   Steve H remarked that these results suggest that it might be more efficient to chop the data up into smaller files.

Steve H mentioned in particular that the netcdf timings could likely be improved by tuning. Mark M asked if the required tuning would depend on observation type, platform, or configuration (in particular the number of MPI tasks), which could make it hard to maintain. Steve H said that it definitely depends on the MPI strategy (for example, currently all MPI tasks read in the entire file and then extract the subset of data they are responsible for), but it's should depend much on obs type.

There were then some questions about the horizontal resolution used for the tests. It was said that c48 corresponds to a resolution of 2 degrees while c192 corresponds to 50km. Some of these test configurations came from Mark O's workflow and he emphasized that these resolutions were specifically chosen to be low in order to maximize the ratio of the time spent for the observations (accessing and processing) over the time spent in computation.

However, Steve H and Steve V emphasized throughout the presentation that neither netcdf4 nor ODB have been optimized for this benchmarking. So, one should not over-interpret the timing results. The purpose of these timing comparisons was to see if there might be dramatic differences in performance that might cause us to exclude one of the options. Such dramatic differences were not found. After we make a decision on which one to use, we will work on optimization.

The situation was similar for file size. For larger files, ODB had a slight efficiency advantage, occupying less memory than netcdf for the same data. For smaller files the opposite was true. After Steve H's presentation the meeting then entered a period of more open discussion.

Ricardo returned to the issue of speed and asked how the numbers on the slides compare to the speed that GSI can read the data from BUFR files. Nobody was quite sure on detailed timings though Ming mentioned times of 1-2 min for large files using 16 MPI tasks. He also mentioned that this is for parallel file reads and Rahul asked if the JEDI implementation takes full advantage of the parallel file access from netcdf4/HDF5. Steve H said no - this does not fully exploit the parallel netcdf4 functionality.

Ricardo said that GSI has two levels of reading operations - this is an inefficiency that should be accounted for when making any comparisons.   Will added that the BUFR files that GSI reads are already thinned and agreed that one must be careful when making comparisons.

With regard to the plot on slide 8 showing "IO Run Time Relative to Total Run Time", Mark O emphasized again that the performance benchmarks were chosen to maximize this number; for typical applications at full resolution, this number is expected to be less than 1%.

Yannick then emphasized that all the science code is the same in the various runs that Steve H presented, independent of the file IO. This alone is a notable take-away point from the presentation, demonstrating that, at least in this sense, we have achieved the Separation of Concerns principle he highlighted earlier.

Will then remarked that the 20% difference in file size (smaller for ODB, see slide 10) is significant and added that scalability is important since file sizes are likely to increase in the future. But, he also added that more could be done to optimize netcdf.

Will then requested a clarification on how the database is populated during an application - do users need to copy over their own versions of the data files or is there a central database that is pre-populated before the application. For example, at GMAO, would there be one copy of the observations that everybody could access. Yannick said yes, when data comes in for an application, it should be just processed and converted to IODA format once. And, there would ideally be one copy of these IODA data files at a particular center that all users can access. One database containing all the observations is appealing but there are access and cost considerations that may prohibit this. For example, if the data base were hosted in the cloud then data egress charges could be substantial. So, model backgrounds and observation files may be stored on different HPC systems as needed. Furthermore, Tom pointed out that systems that are behind firewalls will need local copies of the data. Dick added that a single cloud-based repository of observational data files would be a beneficial service to the community - this is the idea behind R2D2 (see Yannick's second slide). Still, applications at individual centers, particularly secure operational and/or DoD centers, would need their own copies.

Dick then brought the focus back to the principle question of this meeting: what data format should IODA use, netcdf or ODB? He said that the minor efficiency differences reported in Steve H's slides indicate that either option is viable. So, what other criteria could we use to make a decision? The discussion then turned to the functionality, content, and standard/conventions of these two alternatives. In particular, he said that netcdf was initially developed for climate applications whereas ODB was designed from its inception for processing observations within the context of NWP. Will then mentioned a recent workshop that defined standards for NASA observational data files and he mentioned that neither netcdf nor ODB will likely meet those standards - some conversion will likely be necessary.

Rahul then asked if ODB/ODC is available to JEDI users for investigation and testing. Yannick answered that we have a fork of both ODC and Odyssey (pyhon processing tools) on our JCSDA GitHub organization. Mark O added that there is a feature branch of fv3-bundle that uses ODB/ODC.

There was then some inquiries about the licensing. Yannick said ODC will be open source (the first public release is imminent) and will be distributed under the Apache license, the same license used for JEDI.

Ricardo asked that if we were to use ODB, would this provide an opportunity to leverage the ECMWF data store? Dick responded yes - most of the data in the ECMWF data store is publicly available and stored in ODB format. So, it could be ingested directly into JEDI. He emphasized again that OBD has gone through the trouble of coding what is needed to do DA.

Yannick then emphasized that, with our current resource, JEDI can only support one - either netcdf or odc, not both. Dick said that for the end user, the decision should not matter much. Ming asked if we are planning to host a community data hub and Dick responded yes - that is the idea behind R2D2 (though this does not exist yet).

Ming then mentioned that BUFR is 30 yrs old and was developed when computers were relatively slow. Should we work with WMO to develop a new standard? Daryl responded that that is a good idea but there is no chance that this would be implemented in a time frame relevant for JEDI development.

Daryl then said that if it is urgent to make a decision soon, then we are stuck in a corner. From a NOAA standpoint, they cannot make a recommendation until they have had a chance to work with and evaluate both options but ODB has not yet been vetted so they are not able to install it on operational platforms such as WCOSS. Wei-Ya from EMC then said that they have recently hired a new person who could work on the evaluation of ODB/ODC on NOAA computers but it may take a few months before they have access to WCOSS.

Will agrees that it is too soon to make a recommendation until more user/developers have had a chance to work with ODB. He said that a point in favor of netcdf is that is is a standard, the use of which goes beyond the DA/NWP community. For example, it is the standard for satellite data and the underlying hdf5 infrastructure is popular throughout the data science community, with many existing tools that could be used for diagnostics and visualization.

Dick emphasized that the point at issue is what to use for the internal data representation in ioda. Regardless of what we choose, we can always convert to netcdf/hdf5 or other formats for diagnostics and other processing.

Will then asked why we have the ioda layer at all. Dick responded that there are many pre- and post-processing layers in any DA workflow. He referred to slide 3 of Yannick's presentation and said the workflow is just as complex at ECMWF. Ricardo agreed that the workflow will always involve multi-layered data flows.

Daryl added that this is not just a DA/NWP issue - it's much broader and will impact other efforts such as verification. Ricardo said that this is unavoidable - even ECMWF does not have a unified file format across all applications. Daryl disagreed and said that they're using BUFR files now as a unified standard. Yannick added that some effect on downstream users is unavoidable because we not going to use BUFR for JEDI/IODA. Daryl agreed that BUFR is not ideal, but it is not going away. Many users of NOAA data products use BUFR files for a variety of applications. Dick asked if this is by choice - if they had another option, would they use it? He said that there were many users of ECMWF data that had been using BUFR files in the past but wanted something simpler. There was then general discussion that netcdf files are easier to deal with than BUFR files, with more tools to facilitate access. By contrast, BUFR is too specialized but it's the only WMO standard so it's hard to avoid. Daryl warned again that if we were to change the status quo, there will be ramifications.

Ming said there are lessons to be learned from BUFR - WMO declared this as a standard but did not provide adequate tools to work with the format. He added that netcdf has a large community and if we choose that option we should provide support tools. In response to an exchange with Daryl, he added that one problem with BUFR is that it requires external data tables to properly read the files. Dick agreed that this is a problem and added that the data tables can change with time so it is necessary to make sure you have the proper tables when reading a particular set of BUFR files. He said that BUFR is in our world - we have to deal with it. But, it is not a viable candidate for the central question of this meeting, namely the internal data format to adopt for IODA.

Dick then posed the question to the participants of: What do we do next? We cannot continue to support both.

Will said that the next step has to be to begin the vetting process at NOAA and NRL/DoD for ODB/ODC. If they do not approve of its use on operational platforms, then it is a non-starter. At this point, this is more important than the performance comparison.

Steve H agreed. He emphasized again that the performance comparisons have not been optimized so they should viewed with caution. No major bottlenecks were identified so, at this time, the performance results should not be a determining factor. He added that how you organize data within the files is key and this has not yet been optimized. He advocated for making a choice soon so we can make progress with optimizing for one or the other.

Ricardo asked if ODB were only used for internal JEDI DA, then why would EMC/DoD object? Daryl said it is both a security issue and a software support issue. That is why it has to go through a vetting process. Ricardo asked if the same vetting process will be needed for JEDI itself and Daryl responded yes. So, Ricardo suggested that ODC could go through the JEDI vetting process, as a component.

Wei-Ya asked for a deadline - when do we need to make a decision? Tom said there is no hard deadline but the longer we wait, the more it will strain the resources we have; requiring the JEDI team to support two alternatives will pull resources away from other development work.

Daryl said that if a decision is needed right away then it will have to be netcdf since this is already installed on WCOSS. Stan asked if netcdf ever went through a similar vetting process. Daryl said that was before his time but several participants, including Ming, remembered a time more than 7 years ago when netcdf was not approved for NOAA systems and had to be converted to binary files. Ricardo remembered that there was a long discussion involved. Stan recalled that before 7 years ago, WRF was using netcdf formats internally and asked Daryl if a similar approach could be used here. Daryl did not know what the best approach would be but added that this vetting process is critical for operations. Dick said "but this applies to everything we do".

It was then agreed to get the process going, both at EMC and at DoD centers like NRL. Daryl, Dick, and Nancy agreed to meet next week to discuss this further. Tom agreed that this is the right approach - we do not want to be in a situation where we adopt some standard or software for JEDI only to find out later that it cannot be used for operational systems. We need to understand the requirements as soon as possible.

Since we were nearing the end of the allocated time for the meeting, Yannick opened the floor for other questions or comments. Tom had a question for the Navy and Air Force representatives if there is anything else we should consider for those workflows that do not currently use GSI.   Nancy asked what the external requirements might be if we were to adopt ODB/ODC - would it involve a commercial purchase? Yannick responded that it is open-source software with an Apache license so there is no purchase necessary. Nancy said then that the main question would be which one delivers better functionality?

Tom said that from a JEDI perspective, the functionality is equivalent since the IODA API is the same for both. Steve H mentioned there might be a distinction with regard to diagnostics. In trying to think of possible differences, Yannick mentioned that he knows that ODB can handle a continuous DA workflow, such as adding observations on the fly. He wasn't sure if netcdf can handle this. Stan said that is an important consideration for CAM. Ricardo then addressed the UK Met Office - he noted that they use ODB now and asked for their perspective. However, Marek and Steve had left the meeting by that point.

Dick said that the functionality question is an interesting one and that we should check the APIs more thoroughly.

Then there was some discussion about whether ODB/ODC is a database in the technical sense and Yannick and Steve V agreed that it is not - this functionality was previously there but was abandoned for ODC.

Tom asked if the IODA dictionary signal metadata is the same currently and Steve H responded that yes, both use the same table names so this is already transparent to the user.

Yannick then brought the meeting to a close and encouraged all participants to keep the communication going - let us know if you have any further comments, questions, or perspectives.

Tom closed the meeting by expressing that it was very productive and useful and he thanked all for participating.