PANEL ON EXPERIENCE OF USING HIGH PERFORMANCE COMPUTING IN METEOROLOGY

George Mozdzynski, ECMWF

As has become customary at these workshops, the final session was dedicated to a panel where brief statements raising fundamental or controversial issues were made to open the discussion.

Fault Tolerance and Resilience

"I would like hear the experience on resiliency and fault tolerance when running with a large number of cores."

"We have heard from one hardware vendor that resilience cannot be the primary focus of the programmer."

"May be we are not at that level of core count today that we have to worry about this. Obviously as we progress to much higher number of cores we will have to take this into account and find solutions. Until then we expect our systems to just work and not to take any specific action to address fault tolerance in our applications."

"We are trying to solve some specific problems with ensemble forecasting where we have a large number of ensemble members and one approach is to put all these ensemble members into the same executable where historically we have used scripting to provide some level of fault tolerance so that if an ensemble member fails the rest can continue. So we are looking for alternative communication mechanisms to address this particular use case. But as far as general fault tolerance is concerned, that is a very big topic, there are so many ways that things can break."

"If we adopt a master-slave approach of programming we can recognise that a slave doesn't respond within a certain time and requeue that work item to another slave. So we can survive some slave fallout, albeit with some delay. But this means some different thinking than what we do today because today we organise our codes in a very strict manner which is not flexible. To be resilient we would need to have data replicated which would clearly increase the total memory use."

"In the new MPI 3 standard, there is a timeout option on MPI_RECV, so that if you don't receive a response within this timeout period you can assume that particular task/processor is dead and take some action."

"Of course, a number of meteorological centres and HPC users are already doing something about resilience in that they write out restart files for long running experiments, where every few months an experiment fails. Simply requeuing the experiment is sufficient for intermittent failures where execution continues from the last restart file. Clearly this does not apply for failures that are consistently reproducible, where some other action is required, either a namelist change or a fix to the application.”

“In the exascale future we should expect a change in programming paradigm with a symbiotic design where a minor change in hardware design could make software that much simpler and vice-versa.”

“Resiliency and fault tolerance is probably the number two topic today for hardware vendors after power consumption. A lot of work is already being done in this area. Another concern for systems at extreme scale is that of silent errors, where systems are so large that you can get an application that completes normally but can have an error in the data either read or written. How realistic this will be is hard to judge.”

“For resiliency there are two ways to approach it, one is backing up the data to have multiple copies, but another topic that might be investigated is if we go to extreme parallelism and have small patches of data, is there any way to reconstitute enough of the missing data to be able to continue without backup.”

“Could we learn from techniques such as parity checking and raid in disk subsystems to handle the detection of silent errors.”

“Since we have been doing a second level compression of our grib files we have a checksum in these files, and we have encountered that in approximately 20 terabytes of data we have 10 to 15 bit errors after writing to disk and reading the data back again. In some earlier experiments where we downloaded some ERA40 data from the network, we found some files corrupted, which means that there seems to be once in a while some silent errors in disk access that we fail to see and some errors that we only see when one of our application fails.”

“Between storage and memory there is a lot of equipment that data flows through, from controllers to cables and connectors that could have the potential to corrupt data.”

“How can we detect that we have an error, this could be in your initial data, an error in the memory system or communication network. We would hope that today’s systems have some resilience built in with parity checking or something similar for networks, with the ability to retransmit faulty packets and of course reporting this in system logs.”

“I had this experience recently where my application produced different results on about 1 in twenty runs which we tracked down to a single node being the cause. To elimiinate the application being the cause we ran this same application on a different system and that gave identical results on repeated runs. Of course, we would not normally have this possibility of chasing such problems.”

“One approach for data could be to read data back after it has been written and check it is consistent. In the case of a difference you may not know exactly where the problem is.”

“A response we sometimes get from vendors when we suspect a tape drive as the cause of some data corruption, is that there are many of these tape drives being used and nobody is reporting them as having a problem. When we ask vendors for an end-to-end process to check for corruption it seems they are not ready to do this. Maybe the challenge we will face at exascale is something that brings enough revenue to vendors so that they would be more willing to address this."

"I think we need to pay attention to the fact that the problems of silent errors in data are more critical in climate forecasting than in short range forecasting. We have the habit when installing new machines to pay a lot of attention to getting the operational short range forecasts working correctly on these systems, but any errors that occur after this would most likely be 'washed out' by tommorows weather so these errors are less visible. For climate forecasting there is the assumption that the system is working correctly because it is being used for daily forecasts."

"There is one part in forecasting where errors would be persistent and that is in the assimilation cycle. This is a process similar to climate forecasting."

"This maybe a an unpopular thing to say, but isn't it more probable that there is an error in the application."

"If we haven't seen silent errors, it is very possible that the reason for this is because we haven't been looking for them."

"I can recall a problem we had many years ago when we had an XMP-4, which resulted in an occasional failure in our operational forecast. Engineers were called to run exhaustive diagnostics, the machine was run on voltage margins, and no error in the hardware could be found after many weeks or months of effort. Eventually the problem was found, it was an application error, some BUFFER I/O (input) was initiated but was not checked for completion with a UNIT test, which meant that very occasionally data was being used before it had arrived. It was suspected that the frequency of this problem occuring was dependent on what else was running in the system."

"We as a community should make sure that computer vendors understand that we need as much help as possible in error detection. I know our applications are error prone and we need to work hard to eliminate such errors by exhaustive testing. However, as we progress to exascale systems and due the size of these systems parts of the system are going down, it should not be left to us to take care of that, it should be the responsibility of the operating system. We should try to make sure that the useability of the system should not degrade just because they get larger."

"I suspect that solutions to these problems will happen before we get to exascale, as there will be no market for systems with such failures. Whether these failures are handled by the operating system or to some degree by our applications we will find out in the future."

"Many of the vendors are aware of these issues, it is just how we address them with you and take this forward."

Parallel I/O

“One of the biggest inhibitors of scalability occurs when we want to write out some fields at post-processing timesteps. Very often we see an approach where one task is doing the I/O, but it is the data gathering process that it is taking the time. The I/O itself does not seem to be the problem. Maybe there is some semi-generic solution for doing this, not necessarily MPI-IO. We have also seen that memory growth in MPI is killing our applications. If we run with 100K tasks our experience is that MPI is using more memory than our application, this is something that is also being addressed in MPI 3.”

“Something that we are seeing in our code development is a lot of changes to handle I/O. We all need to scale, so parallel I/O is the way to do it, but that requires architecture changes to the code. One approach is to avoid the disk subsystem altogether by sending data to another process to post-process this data.”

“We tried to modify our code to do parallel I/O with MPI-IO. What we found was that this code is very machine dependent . I really hate this. It means we have to redo this every time we change our machine if it has a different architecture. I seems to me that we are lacking standards in our community. The only standard which is available in NWP is grib and a lot of groups don’t use this. The climate community for simplicity often use netCDF.”

“There might be a difference between the forecasting and climate communities in their needs for this data. In the forecast community we need to get the data out in the most efficient way, and it needs to be immediately useable and available to the end user in the format they require and this means it needs to be written as full fields and in grib format. Perhaps in the climate community diagnosis of the data is not done in real time and therefore data can be written out in the most efficient way and processed much later.”

“The problem with the climate community is that the post-processing of the data can take longer than the production of the data. In the end there is a similar problem we are facing to get the data out as efficiently as possible and we should aim to find approaches that are suited to both communities.”

“If one community has a higher demand than the other, then that surely would be the one to use.”

“I think the I/O requirements of forecast and climate communities are mostly the same.”

“Should we have an initiative on I/O issues to prepare for exascale.”

“What are you proposing to standardise, we already have two standards, grib2 and netCDF.”

“I was thinking a lot further. Could we develop something like an in-core database which can be connected by whatever is standard such as TCP/IP4, MPI-2, having clearly defined interfaces.”

“Do we want to do this together? If so, my next question is how.”

“Probably the first step is to find money to do it, the knowledge which is required to do it, and then find a group that could test a lot of possibilities, possibly involving vendors. This is nothing that we should talk more about today, but just to think about this.”

“There are a lot of initiatives at the moment in the area of software for exascale, that could be a potential source for funding.”

“Actually this is not necessary an exascale issue but maybe a current scaling issue.”

“NCAR has their POP parallel I/O framework which would be another reference on this.”

“Do we want to agree on a way of collaborating on this, so that on another occasion when this community meets again we can see progress on this.”

“I think RAPS would be a good framework this, because it could become a useful benchmark that could involve vendors. But to start with, these issues should be initiated in a discussion forum.”

“My proposal would be to set up a discussion forum where we can start by discussing these issues. Then we will see who will take what initiatives to make progress on this.”

“An obvious first step is for the interested parties to summarise how we do I/O today on our systems and what our requirements are. Then put this all together and then maybe we would get a clear view on whether we should be going forward on this or not.”

“We could make it an objective to give a talk on parallel I/O at the NCAR Annecy workshop in Sept 2011.”

What is the requirement for tools from the meteorological community?

“Following the talk that IBM gave, I downloaded the ECLIPSE tools package and it was not that difficult to install compared to two years ago. What about having tutorial on this at the next HPC workshop in two years time.”

“I also spoke to the IBM speaker and he suggested that we might organise a workshop on tools.”

“I think it is worth having a tutorial but this should include more advanced details than is usually done in tutorials.”

“Maybe we should call it a special session within the HPC workshop then think about what should be contained in that session, we should leave it more open for now.”

“My only concern specifically with PHOTRAN IDE, somone should check that the tool is mature enough before we commit to a tutorial. For example, I could take a Java ellipse IDE and push a button and do all sorts of wonderful refactorings and they generally work, but with Fortran that is not supported yet.”

“Maybe we should really try some of these tools with our own codes. Otherwise we could have a situation where the tool works for some small test cases provided as examples, but when we try our large application it fails to work. “

“What tools are we actually talking about testing.”

“I would propose debuggers that work at high core counts.”

“One thing we haven’t discussed are problems that occur at scale that could be an application problem or a system problem, these problems scale with the size of the system, the number and size of the logs generated that have to sent to the vendor, or to be analyzed by us. We definitely need more powerful tools to help us understand the current state of the system, to interact with vendors in a way that doesn’t involve hours of download or upload of logs.”

“It is fairly easy to install something like eclipse on a laptop, but is not so straightforward to install on a system such as ECGATE for 30 users, system adminstrators would have a problem with this.”

“We need tools for I/O, memory and cache use which would access the system’s hardware performance counters. I have seen tools that work with small test cases, but fail with real sized applications.”

“I agree with the previous user, we need tools that work with our main applications, but they should also work across many vendor platforms by have having a common front-end, and small back-ends that interface to specific systems. We don’t want tools that are vendor specific, as we have to learn how to use them each time our system changes.”

“We would also like to see where load-imbalance occurs in our applications.”

“Shouldn’t we compile a list of the tools that we require.”

“Working for a vendor, I would suggest that giving a list to a vendor doesn’t work.”

“I agree, (another vendor) tools are often an afterthought in procurements.”

“I would like to propose we set up a wiki for ideas on tools.”