From Large Simulations to Interactive Numerical Laboratories

Alexander S. Szalay, Johns Hopkins University, Baltimore, MD 21218, USA

High Performance Computing is becoming an instrument in its own right. The largest simulations performed on our supercomputers are now approaching petabytes, and at Exascale everything becomes a Big Data problem. As the volume of these simulations is growing, it is becoming harder to access, analyze and visualize these data. At the same time for a broad community buy-in we need to provide public access to the simulation results. This is only possible if the analyses and visualizations are co-located with the data.

The largest numerical simulations today use tens of millions of hours of CPU time, yet most of the analysis must be performed while the simulations are running, since the output data are too large to be moved or stored for reuse. Scientists in many disciplines would like to compare the results of their experiments to data emerging from numerical simulations based on first principles. This requires not only that we run large simulations and models, but that the results of these are available publicly, through easy-to-use numerical laboratories where anyone can perform their own analyses. Integrating and comparing experiments to simulations is a non-trivial data management challenge. Not every data set from these simulations has the same lifecycle. Some results are just transient and need to be stored for a short period to analyze, while others will become community references, with a useful lifetime of a decade or more. As we have learned over the years, once the data volume reaches a certain size, we have to move the analysis to the data. As our fastest computers reach the exascale, fewer codes will run well on millions of cores, and as a result, fewer people will use these ever larger systems. There will be an increasing gap between the wide science community and these top users. It is increasingly important to create usable science products that can be used by a much broader pool of users; otherwise community support will be soon endangered. It is extremely important to identify a mechanism through which data products from the largest simulations in science can be publicly shared and used, potentially over extended periods.

Data lifecycles of simulations

Most truly large simulations are ***analyzed on the fly,*** analyses are computed while the simulation is running. As these results represent only a small fraction of the data, it is easy to save them to disk. Full restart checkpoints are only generated infrequently. However, if a new analysis idea emerges after the run is completed, the whole simulation needs to be redone.

For ***private reanalysis***, a few tens of snapshots are saved to scratch disks, close to the original supercomputers. An occasional segment of the simulation can be regenerated and reanalyzed from the nearest snapshot. This is typically done by the same team who ran the original simulation.

There are cases when the outputs are made available for ***public reuse***, through sharing a limited number of snapshot files. These are typically placed at a public file server, and can be downloaded. This model of data sharing poses practical limits to data downloads at a few tens terabytes. The limiting factor is the user's network bandwidth, although the available storage at the user's end is also a problem.

Sometimes simulation outputs are made available through ***public service portals***, enabling the users to perform some extractions or computations over the data. This idea of "virtual data" has been around for more than a decade, but it has found limited uses. Creating a public service portal and its functionalities requires a substantial effort, thus it is only worth doing if the data will remain public for an extended period.

There are very few simulations that have reached the stage of ***long term archival*** of their lifecycle. Not every simulation will be equally used by the public. Some will fade into irrelevancy while others emerge widely used. It is these latter which need to be kept for a long time, even if just for comparison and reference.

Petascale numerical laboratories

Even though the largest simulations today are approaching trillions of particles or grid points, the size of the output generated is a few tens of TB, rarely exceeding 100TB and almost never reaching a PB. The larger the computer, the more cumbersome checkpointing becomes, and while the biggest machines have a TB/sec aggregate sequential bandwidth to their storage, copying 100TB will still take 100 sec, too long to do often, limiting the number of snapshots. As the interconnect speeds are not going to increase by a factor of 30-100, it is likely that this limitation remains in place, thus even once we have exascale machines, the outputs will still remain in the few PB range.

For a scalable analysis we need to come up with a data access abstraction or metaphor that is inherently scalable. For the user it should not matter whether the data in the laboratory is a terabyte or a petabyte. The casual user should be able to perform very light-weight analyses, without downloading much software or data. Accessing data through the flat files violates this principle: the user cannot do anything until a very large file has been physically transferred. One can create an immersive environment, where the users can insert ***immersive virtual sensors*** into the simulation, feeding a data stream back to the user. They can provide a one-time measurement, they can be pinned to a physical (Eulerian) location or they can "go with the flow" as comoving Lagrangian particles. In this case, assuming that the sensors can access the data server side very fast, the only scaling is related to the number of particles.

By placing the particles in different geometric configurations, users can accommodate a wide variety of spatial and temporal access patterns. The sensors can feed back data on multiple channels, measuring different fields in the simulation. They can have a variety of operators, like computing the Hessian or Laplacian of a field, or applying various filters and clipping thresholds. This pattern also enables the users to run time backwards, impossible in a direct simulation involving dissipation. Imagine that the snapshots are saved frequently enough that one can interpolate particle velocities smoothly enough. Sensors can back-track their original trajectory and one can see where they came from, all the way back to the initial conditions. We can also imagine cases where one simply retrieves the velocity of the sensor particles, and applies a special equation of motion involving other factors (inertia, friction, stochastic perturbations) and move the particles externally, possibly on a users laptop, anywhere in the world. This simple interface can provide a very flexible, yet powerful way to do science with large data sets from anywhere in the world.

When the primary consideration is checkpoint restart, it is enough to have a small number of snapshots. On the other hand, if the goal is to be able to reconstruct the fine-grained spatial and temporal history of the simulation, and look at any part in detail, it is important to match the high spatial resolution with an appropriate temporal resolution, i.e. a large number of snapshots: the simulations need to be designed and run differently if the target is to create a long-lived numerical laboratory used by hundreds of scientists. There is also a need to create a statistical ***ensemble of simulations***. Instead of focusing on the largest number of particles, or the highest resolution, we also need an ensemble of simulations, which can be used for uncertainty quantification for our various observables.

The Challenges

The premise of this paper is that due to community pressure and demand for repeated posterior analyses some of the best and largest simulations will be turned into public numerical laboratories. This process presents nontrivial challenges for every step of the simulation and reuse process. In particular, we need to figure out (a) where to store the data for the reuse, (b) how to move the data to this location, (c) how to look at it, how to render peta/exabytes, (d) how to interface to the data, (e) what analysis and data access patterns to support, (f) what architectures to use for the posterior analyses.

The long term data storage should use inexpensive storage technology so that it can keep accumulating important data. The Numerical Laboratory has to be interconnected with at a high speed to the primary data source. Most supercomputers are upgrading their connections to 100G. So, while today it is still painful to move data exceeding 100TB, this capability is expected to become increasingly available to a wider set of institutions, while PB-scale data transfers will be possible, but more limited.

The simplest is to provide access to the simulation files. However, these files keep the data in large contiguous chunks, partitioned following the domain decomposition. Such access to the data is only useful if all the data is loaded back into memory. If each snapshot is tens of TB, there are not many user-owned computational facilities that can support such analyses. Having service oriented access with a finer granularity has a lot of advantages: aggregates can be computed and subsets can be extracted on demand. The result is that the user performs the low-level data processing right on top of the data, inside the Laboratory, reducing the volume to be moved across the wide area network. This requires nontrivial computing power, tightly integrated with the data storage. On the other hand, from a community perspective, individual users can run petascale analyses from a laptop, rather than forced to build their own facility; a reasonable value proposition for the community as a whole.

There are several distinct access patterns how a posterior analysis will access the simulation data. ***Global analytics*** require access to the whole simulation volume. Examples of these are computing the Fourier transform of a scalar field, like density, spatial correlation functions, or computing the density for a particle based simulation using a smoothing kernel.

Other analyses are similar to a ***rendering*** of a sub-volume. These include visualizations, but many other computations involving computing projected quantities and maps can be mapped onto rendering hardware. These patterns require accessing a substantial fraction, but not all, of the simulation volume. GPUs can be extremely useful in these computations.

There are computational tasks which require ***localized access***. in order to compute the fluid velocity at a given location, we need to extract a small volume of a few grid cells in each direction, the size of our kernel, and then compute an interpolated value. These patterns require fine-grained localized access to small parts of the data. High-resolution spatial indexes can make a huge difference in enabling fast data reads.

Architectures of numerical laboratories

One might first think that the recent proliferation of commercial cloud solutions will bring an immediate solution to the table. There are several interesting and important aspects of these architectures. The most important is that ***they do not try to be everything for everybody***. Due to their commercial nature they were built with a razor-sharp focus, with major tradeoffs in their design. These are ***extreme scalability, economies of scale and resilience.*** The clouds built by Amazon, Google and Microsoft are all based on the principle that components will fail, systems have to survive and recover, and data cannot be lost. This is accomplished by massive data replication and excessive monitoring. The data in these clouds is largely unstructured, from web content to click streams. The connectivity of the data is mostly large, hard to partition graphs, describing relations or social networks. Much of the software framework for the processing reflects this fact.

Scientific data is quite different. Most of our data, especially the simulations, are highly structured, with specific schemas and data structures. There are well-defined multidimensional domains, with distinct timesteps and boundaries. There are similarities as well, especially how scientists are also exploring phenomenological correlations rather than searching for strict, causal relationships only. Nevertheless, due to the more structured nature of scientific data, the computations can utilize vectorizable architectures better.

The challenge is to define an architecture that has a similar, ***sharp focus for the needs of science***: (i) efficient and inexpensive for storing petabytes of data, (ii) provides high streaming bandwidth to the data, (iii) can occasionally perform compute-intensive tasks involving large amounts of data, (iv) can support additional, more random access patterns. It is clear, that supercomputers have enormous processing power, but they are too expensive for the posterior processing of large data sets. It is also clear that database servers have been successful in providing fast, indexed access to small granularity data, but they do not have the compute power to perform some of the more floating point intensive analyses. Between these two edge cases there is a continuum of problems we must be able to tackle successfully.

Summary

The next generation simulations will have trillions of particles, tens to hundreds of petabytes of output. A wide community of users is ready to use the output of these simulations, and compare them to experiments and observations. The science community has used several of these prototype numerical laboratories successfully, even artfully. It is clear that the stage is set to have a new platform, which bridges the current gap between supercomputers and database servers. It is clear that the key to efficient data analysis is a good data layout and efficient data access. The data access patterns discussed here are quite diverse. Most likely we will need a heterogeneous, possibly hierarchical ecosystem of system components, each optimized (or reconfigurable) for different data access patterns and enough internal bandwidth to move parts of the data to the appropriate system dynamically. This way we can get close to optimal layout for most analysis patterns. In order to do so, we have to understand and characterize how different data access and analysis patterns map onto different system architectures and components.