Supercomputers at Exascale:
BigData and Extreme Computing of the Total Monitoring

Konstantin Stefanov, Vladimir Voevodin
Lomonosov Moscow State University, Research Computing Center

Monitoring systems have become part and parcel of supercomputing software infrastructure. They have traditionally been tasked with controlling hardware and software operations and alerting any emergencies, critical situations and dysfunctions. But the situation has changed radically in recent times. With supercomputing hardware and software becoming increasingly complex and highly parallel, and the efficiency of supercomputing applications remaining low, monitoring systems are now facing a brand new complex objective – maintaining the operational efficiency of supercomputing centers. In this situation, “total monitoring” is no longer a metaphor, but a basic necessity.

This is true today, but the meaning of monitoring for future supercomputers is bound to expand explosively. Thousands of users, applications, computing nodes, processors, accelerators, ports, cables, hardware and software components; many millions of CPU cores, processes, messages; billions upon billions of operations… Ensuring the efficient operation of a supercomputing center requires monitoring absolutely everything that happens inside the supercomputer, and that task alone requires sifting through really big chunks of data. Even for the relatively small “Lomonosov” supercomputer at Moscow State University (1.7 Pflops, 100 racks, 12K nodes, 50K cores), the necessary data arrive at the rate of 120 Mbytes/s (about 30 different metrics analyzed for each node, measured at the frequencies of 0.01-1.00 Hz). It means 3.8 PBytes for 365 days of a year. Importantly, these data have all the traits usually associated with the concept of Big Data: large volume, mixed types, and different sources.

Traditional approaches don’t work very well in this environment. The common approach to analyzing big data is to store the gathered information in a database and then retrieve the numbers required for analysis as it is being performed. This works well when the analysis only requires a small fraction of the overall data volume – for example, if you only need performance analysis for certain selected jobs. But the situation changes drastically when the same approach is used and the analysis requires nearly all of the data produced by the monitoring system. For example, selecting certain jobs based on their performance properties (detecting the jobs with the minimum sustained performance or with the lowest power efficiency) requires the retrieval of all data associated with the given jobs, as soon as each job is completed. As the full set of jobs typically spans the entire supercomputer, all monitoring data originating from the components used to run jobs need to be retrieved and analyzed. This results in the data streams flowing in two opposite directions: one for storing data and the other for retrieving data for analysis. The two simultaneous streams change database access from a cozy linear pattern to something ugly and random. Indeed, while the write stream perfectly fits linear IO, the read stream consists of portions related to each job (additional sampling by nodes and time intervals is needed). These portions cannot be retrieved in large linear reads, and things go from bad to worse when a random read stream is mixed with a linear write stream.

The above reasons plus the fact that most storage systems today are built using HDD’s, or hard disks (which aren’t very efficient for random I/O operations), translate into the need to build increasingly high-spec storage systems for data monitoring purposes. These storage systems turn out to be expensive, yet are only used for data that will be read only once.

A reasonable compromise must be reached on the two key issues: what data needs to be stored in the database, and when should data be analyzed? One way is to store all incoming raw data, then retrieve it as necessary. In this case, we can be certain no data will be lost, and anything we ever need in the future will be available. However, the volume of data will be immense. The monitoring system should be clearly pre-defined; in that case, all challenges inherent in Big Data and Extreme Computing can be overcome by the following principles:

- on-the-fly analysis – all relevant information is extracted from the monitoring data before it's stored in the database;
- on-site analysis – monitoring data must be processed where it is obtained;
- dynamic reconfiguration of monitoring systems – the monitoring system must be capable of adjusting its configuration during the course of its work, depending on the load on the supercomputer and the specific analysis objectives.

Here are a few examples of tasks that can be addressed using monitoring data. One of the most convenient ways to quickly understand the behavior of a particular supercomputer application is JobDigest, which shows not just averaged indicators, but how key parameters change over time (CPULoad, cache misses, flops, number of memory references, Loadavg, IB usage, I/O usage, etc.). JobDigest can be built from raw data stored in the database, but this method will not work with a large number of applications due to excessive load on the disk I/O subsystem. At the same time, building JobDigest won't be a problem, even with a large number of parallel applications, if 1) all relevant information for each application is extracted "on the fly," and 2) the monitoring system is dynamically reconfigured for load balancing by redirecting data flows for building JobDigest to various points in the system. Moreover, if a certain parameter starts oscillating while an application is running, the monitoring system must adapt and, for example, increase the frequency of data collection for that parameter. The situation for collecting integrated system-wide metrics is highly similar: they can be determined from raw data stored in a database, by scanning information on all components of the supercomputer over a certain period of time, or the required analysis can be performed on-site on-the-fly, which requires minimal resources and a correctly configured monitoring system.

A prototype monitoring system based on these principles is currently being tested at the Moscow State University supercomputing center. The results look promising so far. In particular, calculating the average parameters for equipment components over a one-week interval for "Chebyshev" supercomputer (60 Tflops, 625 nodes, 5K cores) originally required about two days. Now, in test mode, it's done on-the-fly for the entire "Lomonosov" petaflops supercomputer with less than 5% load on an auxiliary server (despite the fact that data processing is implemented on the interpreted language Ruby).

In conclusion we would like to emphasize that even a seemingly unmanageable task such as "total monitoring of exascale supercomputers" can be addressed by following a smart approach to reduce "Big Data & Extreme Computing" challenges to a "Reasonable Data & Regular Computing" reality. In this respect, it begs the question: Is Big Data often characteristic of a certain problem, or is it just the result of our misunderstanding of the nature of the problem?..