On the Importance of End-to-end Application Performance Monitoring and Workload Analysis at the Exascale

David Skinner and Alok Choudary

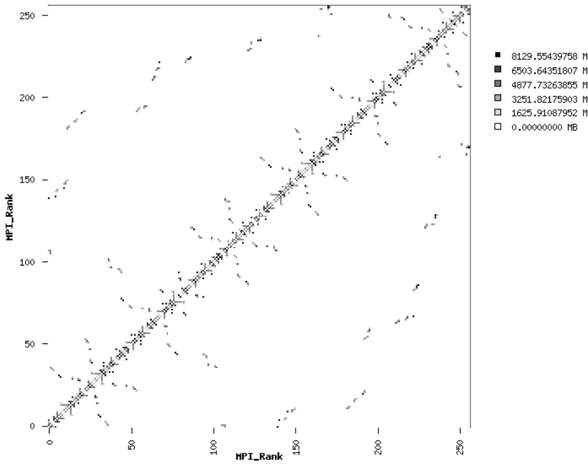
June 21, 2009

This whitepaper sets out to examine the future of performance monitoring on exascale HPC systems. In particular we put forth the idea that such machines will be sufficiently complex that performance monitoring of individual applications and the workload as a whole will change from being a beneficial option to being a necessity. This complexity arises from the number of components and concurrencies expected for such systems. We see the need for a shift from performance monitoring being a useful add-on towards it being a core requirement for basic operation and suggest some beginning steps toward meeting that need.

Performance monitoring has multiple meanings which span a spectrum from lightweight non-invasive performance profiling, to full application tracing, to simulation of applications in virtual environments. For the purposes of this paper we consider only the first context in which production execution of parallel jobs produces a small record that details the applications computation, communication, and memory in aggregate. Metrics such as FLOP rate, memory foot print, percent communication, and message size distributions fall into this category.

On tera and petascale systems, managers and users of HPC systems can benefit in a variety of ways from understanding how a given application interacts with a given architecture. The basic benefit to the user is clear. Having ready access to performance data allows for agile detection of changes in performance due to code or input changes. This same information guides choices about how and where to run, where to look for performance optimizations, and how to estimate the allocation of computer time required to meet research goals.

Taken as an ensemble, performance profiles from multiple simultaneous applications running in a production environment provide the basis for workload analysis. Workload analysis allows HPC facility managers to better procure, provision, and schedule resources to mitigate contention between applications and the architecture. It can also inform of contention between different applications with conflicting resource demands.



It is important to bear in mind that application performance, good, bad, or otherwise is not a property of the application but of the effectiveness in how the application executes on a given architecture. The same notion applies to machine performance. Thus the core importance of workload analysis is a big picture view of the performance ecosystem within and between HPC resources. Such workload analysis can inform HPC managers to make facilities decisions that provide optimal benefit to the scientific community. For example, in an era where interconnect topologies have shrinking levels of connectivity, it is vital to ascertain the impact that will have on scientific applications. One such approach to that workload analysis is to examine the topological degree of connectivity of the applications that make up the workload. The increased FLOP to memory balance brought by many-core architectures brings similar questions to the fore.

Despite the benefits most HPC applications and workloads go unmonitored. A limited and anecdotal view of application performance and workload characteristics was an acceptable situation at terascale, but is becoming problematic at petascale, and will become a critical issue in exascale systems. The impacts of this situation are felt by users who see increasing levels of variability in the runtime of their applications as well as HPC facility staff who are left to unravel increasingly complex performance and failure scenarios on the basis of scant quality data. The costs of not providing a built in approach to performance analysis are beginning to outweigh the costs of operating in the dark. Without an "always on" approach to application performance, petascale performance mysteries will become unfathomable riddles at exascale.

We can draw some useful comparisons from other fields faced with similar complexities. One such example is the ITER, where regular operation of this complex device will require simultaneous simulation to detect problems, confirm observations, and predict optimal operating conditions. Other examples exist in large scale industrial fabrication, sensor networks, and embedded computing spaces.

To borrow a concrete example from chip fabrication spaces, there are certain steps in the fabrication process that once initiated cannot be undone without significant loss and or cost. In those cases it is not unusual to deploy a performance monitoring and prediction infrastructure that monitors power supplies, fans, and host other equipment involved to make an informed go/no-go decision prior to moving forward. Leaning on this analogy a bit further, we point out that time on an exascale computer is likely to be a valuable commodity in its own right. For that reason paying similar attention to performance monitoring is merited over running optimistic that no problems will occur and entering a retrospective mode of performance debugging when problems arise.

Is such application level profiling and workload analysis feasible, especially at exascale? We approach this question by dissecting it and drawing on current experiences at tera and petascale HPC centers. For workload analysis to be feasible application profiling must be feasible. The workload data set is essentially a join on and comparison of the application profiles. For application profiling to be feasible at exascale we must be able to profile codes running on millions of tasks. As long as we make judicious choices about the content and scope of the profiles (see the examples listed above) we believe the data volumes will be manageable with minor R&D effort to scale technologies that have proven themselves at the terascale and are being adopted at the petascale. These approaches aggregate a small set of numbers from each core in each parallel job using the same high-speed interconnect that the application itself uses. The overhead in such approaches is often less than 2% of the overall wallclock time even at the highest concurrencies.

The strategy we suggest makes use of prudent choices about monitoring and scalable methods proven on today's HPC resources. It is lightweight and scalable enough that it can be used in a production setting. It also represents a realization that the HPC community has be somewhat slow in acknowledging, namely that a one-size-fits all approach to performance analysis will not succeed. Asking users or HPC center managers to accomplish their performance analysis needs for production computing at scale will require an approach which is different from, but complimentary to, the more in-depth performance methods embodied in tools which are useful for parallel computing pedagogy and deep dives into code for the purpose of performance debugging.

There are additional motivations for adopting this broad approach to applications and workload performance. As performance feedback auto-tuning becomes more prevalent in application, libraries, and middle-ware the system-wide integrate approach we propose will provide a systematic framework for making good performance decisions within and between codes. Above we mentioned the case of contention between jobs. This is an observed problem on HPC resources leading up to petascale which can have significant impacts on performance. The strategy we suggest would allow for both avoiding the simultaneous scheduling of jobs that conflict with one another as well as the more positive case of symbiotically scheduling portions of the workload that are good matches for one another. Symbiotic scheduling, as opposed the antagonistic case mentioned before, is imagined to become a larger opportunity for effective use of HPC resources as core counts increase. For instance a large number of serial low memory jobs might fit well with a large parallel code that has chosen to idle some cores on each node in the interest of greater memory or interconnect resources per task.

In summary, at exascale all HPC stakeholders will need access to performance profiles of applications and workloads. An approach which does not scale in terms of concurrency or human effort will not meet those goals. We will need an approach to performance analysis that provides data through a method that is as easy as turning on a switch. To the reader that feels this approach is overly pessimistic, we offer the idea that if nothing goes wrong on exascale systems, that is if performance comes out of the box in a consistent reliable way we can always turn off the switch. In the contrary case we should not after the fact scramble to shine light on problems with tools ill suited for the purpose.

Online Analysis and Pattern Discovery of Performance

At exascale level, the traditionally used model of gathering performance data, storing it and subsequently analyzing it, mainly via visualization is not scalable both in terms of overhead and the limited capability in manual discovery. A clear case can be made for automated analysis, mining and discovery of patterns and anomalies in performance of workloads and end-to-end applications. This would require developing data mining driven models for performance prediction and patterns using workloads, and subsequently using these models for performing online analysis of applications using the data generated dynamically while execution. Furthermore, the goal of online analysis would include determination and identifications of anomalies and their causes and using models determination of causes. If bottlenecks and anomalies can be identified online, then they can potentially be used for online mitigation and correction.

Development of workloads for end-to-end workflows:

Traditionally, workloads have been represented as individual applications or application fragments. For exascale systems to be effectively utilized, the entire workflow, as envisioned by an application scientist must be developed as a representative of the workload. This should capture different phases and components of the applications as well as the data movements and interactions between phases of the workflows. Ultimately the richness of these workflows and their ties to scientific outcomes will shape the very concept of performance. A long term goal for HPC performance analysis is to directly connect the inputs and output of architecture choices, application design, and scientific discovery.

1) “ Software Roadmap to Plug and Play Petaflop/s “ William T.C. Kramer, Jonathan Carter, David Skinner, Leonid Oliker, Parry Husbands, Paul Hargrove, John Shalf, Osni Marques, Esmond Ng, Leroy Drummond, and Katherine Yelick, LBNL Technical Report, LBNL-59999, July 2006.
2) “Understanding the Causes of Performance Variability in HPC Workloads”, IEEE International Symposium on Workload Characterization, IISWC05, D. Skinner and W. Kramer