Holistic Performance Analysis for BDEC Systems

– a Big Data Challenge?!

Michael Kluge, Andreas Knüpfer, Ralph Müller-Pfefferkorn, Wolfgang E. Nagel
Technische Universität Dresden
Center for Information Services and High Performance Computing

Extreme-scale computing applications – detailed simulations of natural systems, brain research, analysis of the exploding amount of data in experimentation or sensor networks – require a tremendous increase in computing power. Currently, there is neither a sign for a revolutionary increase of the single core computing power nor for radically new computing concepts providing the necessary steps forward in the near future. So far, the only practical way seems to be evolution: the aggregation of more and more computing and data units following well-established or moderately changed concepts into a single machine.

Even today, the enormous number of computing and data units as well as the complexity of their integration into a single machine provides an intellectual challenge for a human being in understanding and using the full power provided. Hardware hierarchies with different performance specifications, sophisticated network topologies, complex operational software, various levels of (parallel) execution threads, optimization strategies of components that are often unknown and many other implications hinder an efficient use. The progression to millions of units will even amplify these challenges.

State of the art performance analysis tools do a decent job of collecting huge amounts of detailed performance data. They do so by applying massive parallelism on their own behalf. They represent a white box model aimed at completely understanding the reasons behind the observed misbehavior in terms of parallel performance. With that, they try to provide the best insight to solve such misbehavior. Nevertheless, existing systems are restricted either in the detail level of the data collected or in the number of components they get data from. The performance of such a performance analysis is one of the limiting factors by itself.

However, the understanding and hence the energy and cost efficient operation of future BDEC systems requires a holistic performance analysis *(HPA)* approach. It aims at two groups – administrators as well as application developers and users. The former can learn about the system behavior, its performance in interaction with the applications, and its optimization potential. The latter should get information on the application behavior, its overall performance as well as on the various system components, and – finally – optimization hints or automatic self optimization features.

An HPA system needs to continuously collect performance data from all system components. This includes details from all levels of the system:

- “Classical” performance data on the application level
- From the entire computing chain (single core, multi-core, cache levels etc.)
- From the I/O hierarchy – from disks, servers, data and metadata management, client and application I/O requests etc. Applications on BDEC systems will access the I/O subsystem through an I/O middleware. This middleware requires an integrated performance data collection layer.
- From the network components
- About software – versions, parameters, compilation parameters etc.
- From other components affecting the performance

As a result an enormous stream of performance data will emerge. Thus, the collection and the processing of these performance data will itself require the adaptation of Big Data concepts.

Data reduction and efficient I/O are essential – compression, preanalysis, preselection, reduction algorithms etc. combined with directed I/O streaming might even need additional resources for the performance data recording. Data quality methods and classification e.g. by semantic enhancements or an ontology based integration prepare information for the final data analysis. To achieve a holistic performance view on a BDEC system, the final analysis could be either performed post mortem or – depending on the available resources – even online. Knowledge extraction methods from e.g. machine learning, network analysis, ontology based extraction, or other state-of-the-art algorithms will be the base. Modern data visualizations (like multiscale methods) might help but as they need to reduce the complexity of the data to a level realizable for humans their use might be limited.

What insights could such an approach deliver? As stated above, the analysis aims at a holistic view of a BDEC system thus it should correlate all aspects with each other. Thus, it is possible to optimize the performance of the various aspects of such a complex system, e.g. the co-optimization of computing tasks, data access, and message exchange.

An example to achieve this is the analysis and collection of performance patterns. Again, this means patterns on all system levels like

- computing patterns, e.g. the single floating point unit performance
- I/O patterns on all hardware and software layers
- application patterns, e.g. for different parameter or data sets
- compilation patterns, e.g. different compiler flags
- source code patterns/versions

Matching such patterns with the knowledge about and models of the hardware allows answering many performance analysis questions. For instance, a major challenge in the context of I/O will be data driven workflows of heterogeneous software components. This applies not only to parallel applications like large coupled weather models but also includes workflows like MapReduce. On future BDEC systems, these applications need to efficiently utilize the deep hierarchy of storage hardware with different performance characteristics and diverging properties, especially in terms of size, redundancy, and reliability. The access to the storage as a whole will become unsynchronized to a greater extent and more random regarding access sizes. One goal of an I/O performance analysis in these environments is to figure out how to handle data most efficiently on a given architecture. This can be done through the detection of behavioral patterns and their matching with models of the storage hardware.

To efficiently support the users (application developers, system administrators), the analysis must provide precise results of bottlenecks or optimization potential. An example of an optimization hint might be the following: “Function calcMatrix in file matrix.c Version 1.1 running on a processor Intel E5-2690 with an L1 cache size of 64 kB uses 14% of the CPU time of your application. The cache size limits the performance of this function. It might increase by a factor of X if you use a processor with the following cache specifications …”

On a more sophisticated level, which is strongly needed in the mid-term, the system might apply an automatic self tuning. It could optimize the source code according to the used (heterogeneous) computing architecture, use only the hardware/software components that fit best to the applications computing and data specifics, change the data access patterns dependent on the I/O system capabilities, adapt the data flow to the application, or only run applications in parallel that maximize the overall system performance.

Summary:

- The complexity of future BDEC systems requires a holistic view on their performance capabilities.
- Performance data from all levels and units of the system are needed.
- Performance analysis will be a Big Data challenge – Big Data methods need to be applied e.g. for data collection, data quality, or data analysis.
- The results of the analysis must provide the users (application users/developers and administrators) with precise information on optimizations of result in automated self-tuning.