AppSysFusion: CoMingling of appropriate data to drive Codesign of Applications, HPC Platforms, and Monitoring, Analysis, and Feedback Infrastructure

James Brandt, Technical Staff
Sandia National Laboratories
(925) 519-1999, brandt@sandia.gov

Key Points

- **Need to move from considering individual application performance to considering the efficiency of the multi-dimensional HPC ecosystem (i.e., optimize utilization of resources and performance while honoring constraints such as power and priority)**
- **The dynamic and complex nature of HPC workloads requires continuous orchestration of overall HPC ecosystem which relies on continuous insight into all dimensions**
- **We must re-imagine HPC resources as autonomous peer components that can negotiate among themselves and with applications to optimize global efficiency**

Background

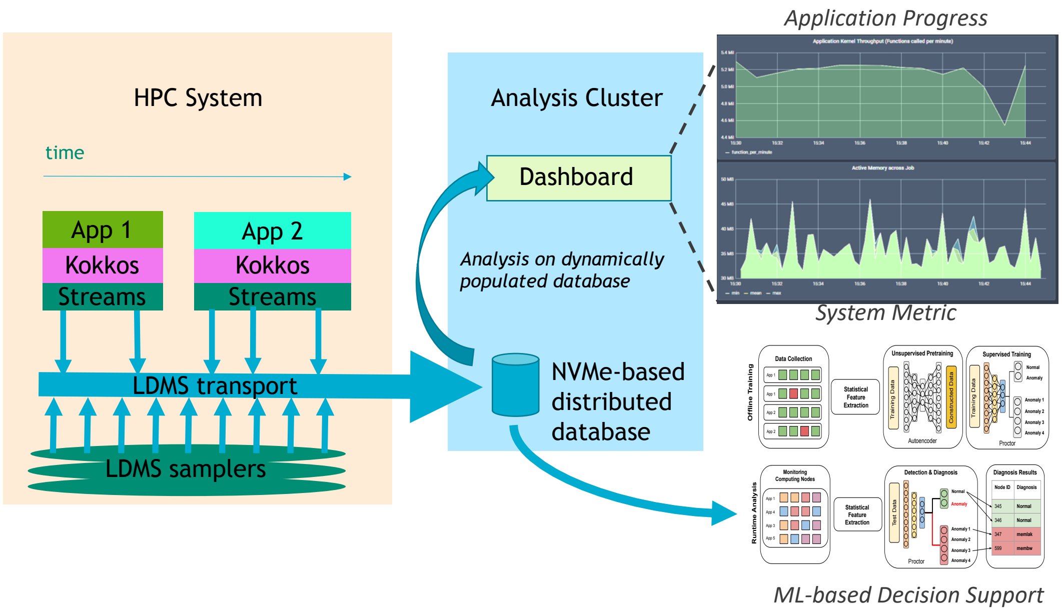
The goal of building HPC systems is to enable execution of large-scale user application workflows in an efficient and performant manner. Performance here is multi-dimensional and includes not just a particular application's time-to-solution, but the aggregate throughput of all applications submitted (workload) and energy spent. The aggregate HPC system power draw must always remain within a contract envelope. Note that workflow in this context refers to an individual user's end-to-end pipeline of executions (some perhaps executed in parallel) and not necessarily a single application run (e.g., iterating a number of times through parameter selection, simulation, results analysis, and visualization). A workload is defined as the combined set of jobs concurrently being executed on an HPC system including flushing of buffers to stable storage and pre-staging of new data to buffers.

Given this efficiency and performance goal, associated constraints, and possible relative prioritization of individual applications and workflows, data-driven scheduling, resource allocation, and feedback will be key to achieving it. Appropriate data from both workflow and system components needs to be acquired and processed in order to make informed resource scheduling and allocation decisions. Additionally, there must be capabilities built into applications and system software to enable them to accept and respond to feedback. Processing of acquired data must be performed on actionable timescales to be of benefit to feedback-enabled applications and system software.

Temporally coherent data collected from across system and workload components is necessary for gaining understanding of how workload components (i.e., individual applications of disparate workflows) contribute to resource load, affect resource performance, and respond to the effects of resource performance changes. As resources become more heterogeneous gaining such understanding (e.g., is it better to execute a particular phase of a workflow on CPU, GPU, FPGA, or? What storage should be used? When, if at all, to checkpoint?) becomes more difficult due to the many options and the need to port applications to these different technologies to obtain technology relevant data. Additionally, understanding how resources respond to oversubscription and how that affects individual execution phases of individual applications, given aggregate resource demand is difficult even in today's workload environment. This environment is only becoming more complex over time as new HPC applications emerge to share the already crowded HPC ecosystem.

Current Approach

To address these challenges, Sandia has launched its AppSysFusion project (shown diagrammatically below). The project comprises functional elements to: 1) collect temporally coherent data with associated absolute timestamp information from all workload components (e.g., application progress/performance measures) and monitorable system elements (e.g., synchronously collect compute node, network, and storage parameters), 2) aggregate collected data to a common distributed data store, 3) process data (e.g., statistical analysis, Machine Learning (ML), generate feedback data) as it arrives to the data store, and 4) provide a visualization portal for users and operations staff to view relevant information.



While AppSysFusion enables collection of application data (via Kokkos depicted above), a consistent method of labeling {application, decomposition, technology, parameter, xxx} combinations is needed to construct and identify models that can be used to allocate "good-fit" resources, ensure resources aren't overloaded, enable automated detection of inefficient behavior, and provide automated feedback to guide more performant/efficient execution/resource utilization. Further, a well-defined information pipeline to enable users and system administrators to evaluate workload, workflow, and resource interactions and utilizations is needed to provide appropriate feedback to application developers and system architects about needed technology directions, instrumentation, and feedback hooks.

Work in Progress

Fundamental changes in resource accounting and management are also required to create the symbiotic relationships and communication paths among workload/workflow components, and HPC resources that can naturally drive performance, efficiency, and throughput toward optimality. Identifying gaps in properties of information and mechanisms for sharing will be needed for co-design of next-generation platforms to further augment hardware and software related performance gains. Challenges to taking this approach are development and testing of: 1) software and hardware to enable scalable distributed decision-making at HPC resources (e.g., spare cycles or dedicated processors on compute nodes, service nodes, and storage devices. Not a dedicated monitoring system) without negatively impacting resources executing user workflow performance, 2) a communication protocol that is lightweight and can accommodate required communication and latency bounds, and 3) system software for birth-to-death management of workflows including addition of self-aware workflow components that can interact in a run time customer-provider relationship with distributed intelligent resource components. Our current approach is to create a container-based emulator in which we can develop and explore interaction of the above-described software components, in the context of a large-scale emulated system, and utilize labeled data traces from production systems as ground-truth for workflow component behavior and resource requirements.

The position paper must include the following Table, which will not count toward the two-page limit. No abbreviations or acronyms should be included in the table.

Table 1: Complete author list and institutional affiliations

Position Paper Author List			**Institution**
Last Name	First Name	Title	Institution Name
Brandt	James	Technical Staff	Sandia National Laboratories
Gentile	Ann	HPC Manager	Sandia National Laboratories

References are optional but recommended

Optional References Here