Ali R. Butt, butta@cs.vt.edu
Virginia Tech

Large-Scale Simulation-Based Resource Scheduling for BigData Frameworks

Ali R. Butt (butta@cs.vt.edu), Chao Wang (chaowang@vt.edu), Kirsh K.R. (kris@cs.vt.edu)
Virginia Tech

The objective of this research is to address the challenges faced in sustaining efficient, high-performance and scalable distributed software frameworks (DSFs), e.g., Hadoop, for supporting data-intensive scientific and enterprise applications on emerging heterogeneous compute, storage and network infrastructure. *We propose to develop and evaluate **Pythia**, an online application-aware oracle framework targeted at DSFs.* A key challenge in designing Pythia for modern applications is that they typically are complex workflows comprising multiple different kernels. The kernels can be diverse, e.g., compute-intensive processing followed by data-intensive visualization, and thus preclude the use of extant static global DSF optimizations. Another problem is faced in evolving DSFs to efficiently handle increasing heterogeneity in the underlying infrastructure. *The novelty of Pythia is that it will be able to adapt to such varying application and infrastructure characteristics at runtime to better drive resource management, consequently achieving high performance and efficiency.*

The infrastructure that supports DSFs is becoming increasingly heterogeneous. One reason for this is that different types of hardware such as CPUs, memory, storage, and network are deployed when large clusters typically go through upgrade phases. However, a more crucial driver for the heterogeneity is the rise of specialized resources — such as FPGAs, GPUs, powerPC, MIPS and ARM based embedded devices and tiered storage. While recent studies have shown that use of specialized accelerators for Hadoop is desirable, sustaining DSFs on such resources is challenging. This is because most modern DSFs are designed to run on homogeneous clusters and cannot effectively handle general purpose workloads on heterogeneous resources. For instance, current Hadoop task slots and straggler detection does not support different core types, e.g., one type of CPU vs. a faster one or a GPU. Moreover DSFs are complex workflow applications with several User Defined Functions (UDFs), which are currently treated as black boxes. An in-depth understanding of the properties of complex workflows is needed to improve the DSF cluster design and resource management strategies. To this end, an application-aware optimizer can exploit a wide spectrum of UDF properties such as data partitioning and functional/algebraic properties, such as monotonicity and commutativity. For example, if the input dataset is sorted, and a transfer function is monotonically increasing, then the output is guaranteed to be sorted as well. If a DSF runtime system can infer this property, it can eliminate a subsequent network hungry data-shuffling phase, therefore significantly improving the runtime performance. Similarly, an optimizer in the workflow execution engine can move a cheap filter ahead of a more expensive operation with which the filter commutes.

The goal of Pythia is to overcome these inefficiencies. The novelty of our approach is that it will be able to adapt to the heterogeneity in applications and infrastructure characteristics at runtime to better drive resource management, consequently achieving high performance and efficiency. Pythia uses a fine-grained simulation of the DSF setups and uses compile-time analysis to learn the application behavior and create a model. At runtime, Pythia implements a heuristics modeling engine, e.g., for MapReduce, which uses the application model, the specifications (e.g., node and network configurations, data layout) of

the resources to be allocated to the application, and predicts expected application performance. Pythia integrates the simulation with DSF runtimes to provide online performance and system behavior predictions. The current system states from live DSF instances, as well as the predictions, are then employed to match the applications with appropriate computing and data components. Such resource--application matching will also support the use of specialized resources such as accelerators and tiered storage. Thus, Pythia will provide a comprehensive solution for realizing efficient DSF deployments that automatically manage heterogeneous infrastructure using application-specific attributes derived from both compile-time and runtime analysis, features that are absent from the state of the art.

Evaluation plan:

The work will focus on designing an online oracle to guide Hadoop resource management. During the course of our investigation we will leverage existing open-source software and solutions where available. The research will require access to large heterogeneous clusters including specialized resources such as FPGAs, GPUs, ARM based embedded devices, and high-end server-on-chip solutions, suitable for testing the proposed tools. These will be used for testing and evaluation of Pythia under the discussed scenarios.

A hurdle in this exploration is the availability of large testbed, similar to what is being designed under the *Cloudlab* project. For workflow analysis, we plan to leverage our earlier work to generate and synthesize workloads. Such synthesis will take into account factors such as job length distribution, job and data dependencies, and computation-IO ratio, and reflect the realistic cloud workloads. Access to large representative testbeds will enable us to fine-tune Pythia design and effect optimizations that can be used in real deployments and data centers. We will also collect data from such deployments, as well as leverage industry contacts to obtain workload traces and feedback on our design. Pythia trades off a highly-accurate but slow simulation with a much-faster but slightly less accurate modeling. However, the novelty is that, since our approach is dynamic, we are able to periodically update the simulation parameters from the actual application run (using these metrics) on each scheduling epoch. This feedback loop prevents the simulation-based predictions from diverging significantly from the actual execution, and yield overall accurate and fast predictions. Testing Pythia on datacenter-scale testbeds offered by Cloudlab will benefit these aspects of our design.

Moreover, these metrics will help to determine the efficacy of Pythia, as well as the ability of our tools to manage heterogeneity in large deployments. Moreover, access to a large-scale testbed will enable us to measure metrics such as workflow turn-around time, job execution time, network and disk bandwidth consumption, and locality for our data and task placement strategies, which can enable us to fine-tune the DSFs. Furthermore, we will evaluate and validate the impact of our design decisions on application performance. Finally, the Pythia software artifacts will be fine-tuned and updated based on the experimental findings and observations in large heterogeneous deployments.

The Cloudlab resources will enable us to better achieve the goals of Pythia by enabling testing and evaluations on realistic heterogeneous clusters at scale.