Accelerating Scientific Discovery through Data-Driven Control and Scalable Real-Time Analytics

Ben Ring, Yanif Ahmad, Tom Woolf. The Johns Hopkins University. (Contact: Yanif Ahmad, yanif@cs.jhu.edu)

We argue for data-driven control as an approach to adaptive computation in the context of *large-scale data exploration* in science and engineering applications. Here, a streaming scientific analysis workflow is continuously tuned with a decoupled, semi-autonomous control task to maximize domain-specific analysis objectives. Data stream systems and big data systems increasingly support iterative computations for machine learning and analytics tasks that repeatedly modify a set of parameters (e.g., convex optimization or matrix factorization) as part of their computation. However, while both adaptive computations and iterative computations build on cyclic dataflows and parameter updates, current iterative computation abstractions are poorly suited to the requirements of adaptive and steered computation: Their cyclic dataflows and parameter modifications are tightly coupled to the algorithm of interest, supporting only a limited degree of adaptability and elasticity on resources. In this whitepaper, we outline a series of systems design challenges for data-driven control motivated by large-scale simulation science, and with broader potential applicability to mobile healthcare, and sensor networks.

Motivating application: adaptive control of molecular dynamics simulations. Large-scale scientific simulations use national supercomputing resources to generate high-resolution datasets for use in exploring complex scientific processes. Molecular dynamics simulations that generate molecular trajectories are widely used in biophysics, and biochemistry for conformation and protein dynamics analysis. These analyses often seek rare events (e.g., transitions in state space) that only occur at long simulation time scales. As such, these simulations are often launched as long-running jobs, using parameter sweeps to study pertinent configurations. This is a natural application for data-driven control: We aim to leverage streaming analysis of the simulation data to intelligently decompose, and manage simulation tasks. By breaking up large sweeping simulation tasks into fine-grained units, we can perform data exploration on-the-fly at the supercomputer, and at the same time, substantially reduce the data movement requirements that impede the use of these datasets outside supercomputer environments.

Motivating application: adaptive sourcing in metabolic syndrome monitoring. With advances in low-cost medical monitoring devices, and lab testing, continuous primary healthcare workflows are rapidly emerging as a high-impact streaming application. Our group is actively developing an iPhone application based on Apple's ResearchKit and HealthKit frameworks to collect and analyze human circadian rhythms spanning eating, exercise and sleep activities. Our focus is on understanding metabolic syndrome in a large-scale population, and correlations and impacts from the temporal relationships in the aforementioned activities. Here, data-driven control can be used to adaptively source sensing data from users' iPhone devices, to collect data pertinent to exploration tasks and hypothesis tests, while minimizing energy expended on communication and querying against population-wide aggregates pushed to devices.

Challenges and Design Goals in Data-Driven Control Systems:

1. *New abstractions for autonomous, elastic and interruptible computation.* As inspired by the supercomputing and mobile settings, data-driven control is often necessary in ad-hoc computing environments, where we cannot deploy daemonized software infrastructure as commonly used in databases, streaming and big data frameworks. For example, supercomputers typically use scheduler frameworks such as SLURM, PBS or SGE to launch jobs, and while parallel batch jobs can be used to instantiate HDFS, Hadoop or Spark, such deployments are typically at a much smaller scale than on cloud deployments due to the waiting times involved in simultaneously acquiring large numbers of nodes. Instead, we argue that *serverless* abstractions as commonly found in embedded databases such as SQLite are necessary, where *no critical system component* is implemented as a long-running process system state in memory. We propose a *self-managing macrothread* abstraction in place of the standard operator-centric abstraction of data systems, to enable the specification of a high-level, autonomous, iterative unit of computation that maintains an efficient, persistent (on-disk) representation of its state. A macrothread is comprised of multiple subtasks that execute in parallel, with

elastic growth and reduction in its resource usage through an external scheduler framework. In our simulation application, by implementing each workflow task (simulation, analysis, querying, optimization and control) as separate macrothreads, our framework is fully interruptible, enabling a maximal ad-hoc usage of available resources for each task. Given their design for repeated ad-hoc computations, we envisage that this abstraction will also be beneficial for programming cloud-based workflows that can take advantage of spare resources such as EC2 spot instances.

2. *Joint optimization of control and analysis.* A key design principle of data-driven control is to exploit both analysis outcomes and resource availability in performing workflow tuning. For example, in our molecular dynamics application, simulation tasks that generate uninteresting analysis results are given a low execution priority. The challenge here is to couple analysis and control with low overheads, and to this end, we consider multi-resolution analysis techniques that realize multiple levels of dimensionality reduction for both exploration and control purposes. Our cost-based control objective balances the convergence of our analyses at each resolution, while using the coarsest resolution for adaptively managing candidate simulation runs.

3. *Declarative programming of control systems, and compiling specialized data workflows.* While macrothreads are the building blocks of our data-driven control framework, we see a need to provide control-oriented programming abstractions that target these macrothreads. By viewing control systems as specifying a user-defined objective and (soft) operational constraints in a declarative rule-based language (e.g., Datalog), one challenge is to synthesize and compile the controller macrothread and its interaction with analysis tasks. This must occur alongside other mechanisms that define dataflows based on the inputs and outputs of each macrothread in a data-driven control system.

   In addition to the steering and adaptive computation that are the immediate focus of this whitepaper, our group's broader interests lie in declarative systems programming for big data applications. To this end, we have developed the K3 big data systems framework (https://github.com/damsl/k3), which comprises an event-driven programming language, compiler, and runtime for building novel distributed data systems, and distributed shared memory abstractions. Our experiments with K3 on a range of analytics database benchmarks (TPC-H, and Amplab Big Data Benchmark), and machine learning queries show that K3 outperforms Apache Spark (a popular generic compute platform) and Cloudera Impala (an open-source distributed database) by 2.6x-74x and 1.8x-56x respectively at scales up to 250GB deployments across 256 cores. K3's research focus is on achieving memory-efficient processing in distributed functional dataflows as popularized by the MapReduce paradigm. In particular, K3 provides a distributed functional and actor-based programming paradigm while supporting consistent in-place updates with the use of program effects and lineage analysis techniques. Furthermore, K3 provides a specialization framework that allows systems developers to customize query and engine logic, in addition to co-optimizing their functionality through the use of aspect-oriented generative programming techniques.

Team profile: Yanif Ahmad is an Assistant Professor of Computer Science at the Johns Hopkins University. In addition to K3 and data-driven control systems, he has substantial expertise in stream systems having been a lead developer on the Borealis distributed stream processing framework, and the DBToaster (http://www.dbtoaster.org) incremental view engine for update stream processing. Tom Woolf is a Professor of Physiology at the Johns Hopkins School of Medicine and holds a secondary appointment in the Computer Science Department. Tom and Yanif have been collaborating on the Molecular Dynamics Database project as a proof-of-concept data-driven control framework for conformational analysis and computational drug design, as well as the metabolic syndrome application described herein. Ben Ring is a Computer Science PhD student at JHU, working with Yanif on data-driven control frameworks. Ben is a Lieutenant Colonel in the US Army pursuing his PhD degree under an NSA Fellowship, with a background in operational logistics and information technology in military applications.