Data Dwarfs: Motivating a Coverage Set for Future Large Data Center Workloads

Mehul Shah, Parthasarathy Ranganathan, Jichuan Chang, Niraj Tolia, David Roberts, Trevor Mudge

HP Laboratories
HPL-2010-115

Keyword(s):

Emerging workloads, data center, data centriCity, coverage set

Abstract:

Recent trends in systems architecture include the growing importance of warehouse-sized computers and new solutions to address the scalability and power efficiency challenges in such large scale data centers. The key drivers behind this rapid growth are a new class of large-scale applications that constantly push the capacity and capability of existing infrastructures to the limit. The essence of these applications is distributed processing of large datasets to satisfy multi-dimensional service-level requirements. A key need for further research from the broader community on architectural issues for such large-scale data centers is the availability of a representative set of the emerging distributed workloads that drive these markets. This paper discusses this challenge. Specifically, we recognize the data-centricity of these workloads and discuss changing requirements in the context of these workloads. We discuss a data-centric workload taxonomy that seeks to separate the most important dimensions across which these applications differ. By examining existing and emerging workloads, we argue for a systematic approach to derive a coverage set of workloads based on this taxonomy. Inspired by the "seven dwarfs" of numerical computation [1][2], we believe that our community needs to collectively identify a set of "data dwarfs" or key data processing kernels -- that provide current and future coverage to this space and can be modeled by open benchmarks with realistic datasets -- for reasoning about new architectural designs and tradeoffs. This discussion was initiated at the 2010 ACLD workshop and we hope such goals would be achieved together by the computer architecture community.

External Posting Date: November 8, 2010 [Fulltext] Approved for External Publication
Internal Posting Date: November 8, 2010 [Fulltext]
To be published in 2010 Workshop on Architectural concerns in Large Datacenters (ACLD) and SIGARCH CAN

Data Dwarfs: Motivating a Coverage Set for Future Large Data Center Workloads

Mehul Shah, Parthasarathy Ranganathan, Jichuan Chang, Niraj Tolia; HP Labs, Palo Alto, CA
David Roberts, Trevor Mudge; University of Michigan, Ann Arbor, MI
{mehul.shah,partha.ranganathan,jichuan.chang,niraj.tolia}@hp.com, {daverobe,tnm}@umich.edu

1. MOTIVATION

Recent trends in systems architecture include the growing importance of warehouse-sized computers and new solutions to address the scalability and power efficiency challenges in such large scale data centers. The key drivers behind this rapid growth are a new class of large-scale applications that constantly push the capacity and capability of existing infrastructures to the limit. The essence of these applications is distributed processing of large datasets to satisfy multi-dimensional service-level requirements.

A key need for further research from the broader community on architectural issues for such large-scale data centers is the availability of a representative set of the emerging distributed workloads that drive these markets. This paper discusses this challenge. Specifically, we recognize the data-centricity of these workloads and discuss changing requirements in the context of these workloads. We discuss a data-centric workload taxonomy that seeks to separate the most important dimensions across which these applications differ. By examining existing and emerging workloads, we argue for a systematic approach to derive a *coverage set of workloads* based on this taxonomy. Inspired by the "seven dwarfs" of numerical computation [1][2], we believe that our community needs to collectively identify a set of "data dwarfs" or key data processing kernels — that provide current and future coverage to this space and can be modeled by open benchmarks with realistic datasets — for reasoning about new architectural designs and tradeoffs. This discussion was initiated at the 2010 ACLD workshop and we hope such goals would be achieved together by the computer architecture community.

2. DATA CENTRICITY

Prior reports (e.g., [3]) have observed the increasing intensity of data processing in data centers, quoting the sheer amount of data to be processed and the mismatch between conventional architectures and the need to quickly access large datasets. Extrapolating these trends, the future will see a greater shift, from compute, to data. Data will likely be the determining force in the data centers triggering different kinds of computation, as opposed to the traditional model of computation that transforms data from input to output. This has been demonstrated in the evolution of both web services and HPC applications. For example, web query and click streams not only initiate conventional searches and online transactions, but also trigger mash-ups and personalized recommendations, and provide the basis for trend and anomaly detection, data mining, knowledge extraction and future prediction. Conventional scientific simulations are moving into data-derived models, data mining, and multiple interactive models that require new balanced architecture designs [4] – a trend that Jim Gray referred to as "eScience".

These emerging data-centric workloads have some interesting characteristics that differentiate them from prior workloads.

1. Scale. Emerging data-centric workloads involve complex analysis at an immense scale. The scale is reflected in the fact that the total amount of data involved in a single operation often exceeds a single-system's capacity, in turn, requiring distributed infrastructure to host and process the data. More interestingly, while Moore's law is a powerful exponential curve in itself, data volume rides on a much steeper growth curve than even Moore's law. For example, online data and enterprise data warehouse sizes have been tracked to increase by more than 3X every two years [5][6]. These trends imply growing system sizes determined by the datasets, and a corresponding need to balance the computing and communication around these increased storage needs.

2. Integration and correlation over multiple data sources. The data growth is partly fueled by the introduction of new data sources (e.g., sensors and digitization of our physical world), but also the integration and cross correlation of multiple data sources (e.g., mash-ups and multi-model interactions). This implies that future data centers are likely to process unstructured, structured, and rich media data as well as their combinations, with diverse data transformation and presentation requirements.

3. Time criticality. A key aspect of data-centric computing is to deliver the right information at the right time. This often translates into real-time or interactive response requirements from the data center, and needs aggressive filtering and summarization as well as architectural support for providing large in-memory processing.

4. Complex mining and learning. Deep analysis, mining and learning algorithms are needed to extract meaning out of the huge dataset. Combined with simple and predictable pre-processing tasks, this will create a spectrum of data processing tasks with varying compute complexity and data access patterns. In contrast to traditional data processing operations like ingress/egress, or simple joins, future processing will focus on more complex operations like cubing, graph traversal, etc.

3. A DATA-CENTRIC TAXONOMY

Clearly, as a community we need a benchmark suite to represent future data-centric workloads. However, existing benchmarks, including recent web2.0 benchmarks [7][8], are not sufficient in capturing the multifaceted requirements and continuous evolution of this domain. In this paper, we argue that maybe, we should consider a systematic approach towards a coverage benchmark set. The coverage should be tested against a well-defined data-centric taxonomy, and, ideally, a small set of key data processing kernels constitute the benchmark suite whose fundamental behaviors persist along the paths of workload evolution.

Table 1 illustrates a data-centric taxonomy based on our examination of a wide class of data-centric workloads. Around the notion of data-centricity, we qualitatively identify important dimensions under which a given workload could be categorized. These include response time (real-time vs. background), access pattern (random, sequential or permutation), working set (all vs. partial), data type (structured, unstructured and rich media), read/write, and processing complexity (low, medium or high). Notice that scale is assumed for all workloads and therefore not explicitly listed. Table 1 also explains the attributes of each dimension and provides a list of popular workloads at the end.

Response Time	Real-time	Real-time or interactive responses required
	Background	Response time is not critical for user needs
Access Pattern	Random	Unpredictable access to regions of data store
	Sequential	Sequential access of data chunks
	Permutation	Data is re-distributed across the system
Working Set	All	The entire dataset is accessed
	Partial	Only a subset of data is accessed
Data Type	Structured	Metadata/schema/type are used for data records
	Unstructured	No explicit data structure, e.g., text/binary files
	Rich media	Audio/video and image data with inherent structures and specific processing algorithms
Read vs. Write	Read heavy	Data reads are significant for processing
	Write heavy	Data writes are significant for processing
Processing Complexity	High	Complex processing of data is required per data item. Examples: video trans-coding, classification, prediction
	Medium	Simple processing is required per data item. Examples: pattern matching, search, encryption.
	Low	Workloads dominated by data access with few compute operations. Examples: sort, upload, download, filtering, and aggregation.
Popular workloads		
Photograph processing, Sensor networks, Web search, Ad-hoc queries, Personalization, Recommendation, BI analytics, Online games, Graph mining, Social network analysis, Ad analysis, Disease outbreak prediction, Media trans-coding, Transaction processing, RMS, Web server, Data mining, Sorting, Decision support, De-duplication, Mash-ups, Summarization, Compression, Encryption, Song recognition, Aggregation, Correlation, Index building, Cubes		

Table 1: A data-centric workload taxonomy

4. AN EXAMPLE COVERAGE SET

Table 2 shows an example of mapping some popular workloads from prior studies to the taxonomy and picking a small subset with full coverage. Here each row represents a workload, and each column is an attribute in the taxonomy. An "X" sign at the intersection indicates the workload demonstrates the corresponding attribute. The highlighted workloads constitute a subset that collectively covers all attributes in the taxonomy.

	Access: Random	Access: Sequential	Access: Permute	Time: Real-time	Time: Background	Compute: High	Compute: Low	Write-Heavy	Read-Heavy	Working Set: All	Working Set: Partial	Data: Structured	Data: Unstructured	Data: Rich media
Sort			X		X		X	X		X		X		
Search (web)	X			X		X			X	X		X		
Search (image)	X			X		X			X	X			X	X
Search Indexer	X				X	X			X	X		X	X	X
Recommender	X				X	X			X	X		X		
De-duplication		X			X		X		X	X		X	X	
Transaction	X	X	X	X			X	X			X	X		
Decision Support	X	X	X		X	X			X		X	X		
Video Sharing	X	X		X	X	X			X		X	X	X	X
Data Mining			X		X		X		X	X		X		

Table 2: An example coverage set of benchmarks

For this set of popular workloads, Table 2 shows how we can choose five workloads to provide a reasonable coverage set while also enabling systems studies including potentially simulation. (1) **Sort** models a two-pass distributed sort at petabytes scale. It is both read- and write-heavy, and stresses the balance between compute/storage/networking subsystems. (2) **Search** models text search using in-memory index to achieve sub-seconds response time. It is read-only and stresses random access pattern. (3) **Recommender** represents parallel machine learning algorithms (e.g., for making Netflix movie recommendations), which have high processing complexity and regular communication patterns. (4) **Dedup** implements mostly read-only, sequential access based data de-duplication. (5) **Video** models a video upload and streaming server with real-time user interaction requirements.

5. GENERALIZATION: DATA DWARFS

Notice that the highlighted subset in Table 2 is neither the minimal set nor the only coverage set. There are other reasons for us to choose them, e.g., scalability and ease of simulation. More importantly, these seem to imply to the potential of identifying a few key "data dwarfs" — data processing kernels that represent critical application classes whose high-level characteristics are likely to persist across implementations and architectures in the future. We argue that such data dwarfs are likely to be well matched for systems/architecture research, for us to understand, design and evaluate data-centric workloads and systems, and are likely to differ from existing compute dwarfs [2].

We also speculate that a small set of data dwarfs will constitute the coverage set in our data-centric taxonomy. Empirically, this seems evident in most data processing frameworks: the majority of database cycles are spent on a handful of relational algebraic operators (e.g., selection and join), and MapReduce frameworks mainly operate on sort, shuffle, serialization and compression.

6. CONCLUDING REMARKS

In this paper, we take a first step towards motivating and defining a coverage set of future data center workloads with a data-centric workload taxonomy and "data dwarfs". Many open questions remain to be answered: Are there other aspects of data-centricity? What new dimensions and attributes should be added to the taxonomy? Where do we find data dwarfs? What are the data dwarfs? How do we model these data dwarfs through open benchmarks and more importantly, open datasets? We hope these questions can fuel the discussion between architects and data center experts, and help collect community inputs to pursue the much needed data-centric benchmark suite.

7. REFERENCES

[1] P. Colella, Defining Software Requirements for Scientific Computing. DARPA presentation, 2004.

[2] K. Asanovic, et al. The Landscape of Parallel Computing Research: A View from Berkeley. UC-Berkeley EECS TR 2006-183. 2006.

[3] R. Bryant. Data-Intensive Supercomputing: The case for DISC. CMU Technical Report CMU-CS-07-128. 2007.

[4] V. Sarkar, W. Harrod, and A. Snavely. Software Challenges in Extreme Scale Systems. SciDAC, 2009.

[5] M. Mayer. Innovation at Google: the Physics of Data. PARC Talk. http://www.parc.com/event/936/innovation-at-google.html

[6] R. Winter. Why are Data Warehouses Growing so Fast? http://www.b-eye-network.com/view/7188.

[7] K. Lim, et al. Understanding and Designing New Server Architectures for Emerging Warehouse-Computing Environments, ISCA 2008.

[8] A. Fox, et al. Cloudstone: Multi-Platform, Multi-Language Benchmark and Measurement Tools for Web 2.0. CCA08, 2008.