RC25281 Revised (AUS1303-002) March 7, 2013
Computer Science

IBM Research Report

Understanding Systems and Architecture for Big Data

William M. Buros[1], Guan Cheng Chen[2], Mei-Mei Fu[3], Anne E. Gattiker[4], Fadi H. Gebara[4], Ahmed Gheith[4], H. Peter Hofstee[4], Damir A. Jamsek[4], Thomas G. Lendacky[1], Jian Li[4], Yan Li[2], John S. Poelman[3], Steven Pratt[1], Ju Wei Shi[2], Evan Speight[4], Peter W. Wong[1]

[1]IBM STG
11501 Burnet Road
Austin, TX 78758
USA

[2]IBM Research Division
China Research Laboratory
Building 19, Zhouguancun Software Park
8 Dongbeiwang West Road, Haidian District
Beijing, 100193
P.R.China

[3]IBM SWG
555 Bailey Avenue
San Jose, CA 95141-1003

[4]IBM Research Division
Austin Research Laboratory
11501 Burnet Road
Austin, TX 78758
USA

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Understanding Systems and Architectures for Big Data

William M. Buros, Guan Cheng Chen, Mei-Mei Fu, Anne E. Gattiker, Fadi H. Gebara, Ahmed Gheith, H. Peter Hofstee, Damir A. Jamsek, Thomas G. Lendacky, Jian Li, Yan Li, John S. Poelman, Steven Pratt, Ju Wei Shi, Evan Speight, Peter W. Wong

International Business Machines Corp.

{wmb,mfu,gattiker,fhgebara,ahmedg,hofstee,jamsek,toml, jianli,poelman,slpratt,speight,wpeter}@us.ibm.com
{chengc,liyancrl,jwshi}@cn.ibm.com

ABSTRACT

The use of Big Data underpins critical activities in all sectors of our society. Achieving the full transformative potential of Big Data in this increasingly digital world requires both new data analysis algorithms and a new class of systems to handle the dramatic data growth, the demand to integrate structured and unstructured data analytics, and the increasing computing needs of massive-scale analytics. In this paper, we discuss several Big Data research activities at IBM Research: (1) Big Data benchmarking and methodology; (2) workload optimized systems for Big Data; and (3) a case study of Big Data workloads on IBM Power systems. Our case study shows that preliminary infrastructure tuning results in sorting 1TB data in 8 minutes[1] on 10 PowerLinux™ 7R2 with POWER7+ systems [5] running IBM InfoSphere BigInsights™. This translates to sorting 12.8GB/node/minute for the IO intensive sort benchmark. We also show that 4 PowerLinux 7R2 with POWER7+ nodes can sort 1TB input with around 21 minutes. Further improvements are expected as we continue full-stack optimizations on both IBM software and hardware.

1. INTRODUCTION

The term "Big Data" refers to the continuing massive expansion in the data *volume* and *variety* as well as the *velocity* and *veracity* of data processing [12]. Volume refers to the scale of the data and processing needs. Whether for data at rest or in motion, i.e., being a repository of information or a stream, the desire for high speed is constant, hence the notion of velocity. Variety indicates that Big Data may be structured in many different ways, and techniques are needed to understand and process such variation. Veracity refers to the quality of the information in the face of data uncertainty from many different places: the data itself, sensor precision, model approximation, or inherent process uncertainty. Emerging social media also brings in new types of data uncertainty such as rumors, lies, falsehoods and wishful thinking. We believe the *4 Vs* presently capture the main characteristics of Big Data.

The importance of Big Data in today's society can not be underestimated, and proper understanding and use of this important resource will require both new data analysis algorithms and new systems to handle the dramatic data growth, the demand to integrate structured and unstructured data analytics, and the increasing computing needs of massive-scale analytics. As a result, there exists much research in critical aspects of emerging analytics systems for Big Data. Examples of current research in this area include: processor, memory, and system architectures for data analysis; benchmarks, metrics, and workload characterization for analytics and data-intensive computing; debugging and performance analysis tools for analytics and data-intensive computing; accelerators for analytics and data-intensive computing; implications of data analytics to mobile and embedded systems; energy efficiency and energy-efficient designs for analytics; availability, fault tolerance and recovery issues; scalable system and network designs for high concurrency or high bandwidth data streaming; data management and analytics for vast amounts of unstructured data; evaluation tools, methodologies and workload synthesis; OS, distributed systems and system management support; and MapReduce and other processing paradigms and algorithms for analytics.

This short paper briefly discusses three topics that IBM researchers and colleagues from other IBM divisions have been working on:

- Big Data benchmarking and methodology
- Workload optimized systems for Big Data
- A case study of a Big Data workload on IBM Power systems

We highlight the research directions that we are pursuing in this paper. More technical details and progress updates will be covered in several papers in an upcoming issue of the IBM Journal of Research and Development.

[1] All performance data contained in this publication was obtained in the specific operating environment and under the conditions described below and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

Contact Author: Jian Li, IBM Research in Austin, 11501 Burnet Road, Austin, TX 78758; Phone: (512)286-7228; Email: jianli@us.ibm.com
IBM Research Technical Report. July 10, 2012.

2. BENCHMARKING METHODOLOGY

Massive Scale Analytics is representative of a new class of workloads that justifies a re-thinking of how computing systems should be optimized. We start by tackling the problem of the absence of a set of benchmarks that system hardware designers can use to measure the quality of their designs and that customers can use to evaluate competing hardware offerings in this fast-growing and still rapidly-changing market. Existing benchmarks, such as HiBench [11], fall short in terms of both scale and relevance. We conceive a methodology for peta-scale data-size benchmark creation that includes representative Big Data workloads and can be used as a driver of total system performance, with demands balanced across storage, network and computation. Creating such a benchmark requires meeting unique challenges associated with the data size and its unstructured nature. To be useful, the benchmark also needs to be generic enough to be accepted by the community at large. We also observe unique challenges associated with massive scale analytics benchmark creation, along with a specific benchmark we have created to meet them.

The first consequence of the massive scale of the data is that the benchmark must be descriptive, rather than prescriptive. In other words, our proposed benchmark is provided as instructions for acquiring the required data and processing it, rather than providing benchmark code to run on supplied data. We propose the use of existing large datasets, such as the 25TB ClueWeb09 dataset [9] and the over 200TB Stanford WebBase repository [10]. Challenges of using such real-world large datasets include physical data delivery (e.g., via shipped disk drives), and data formatting/"cleaning" of the data to allow robust processing.

We propose compute- and data-intensive processing tasks that are representative of key massive-scale analytics workloads to be applied to this unstructured data. These tasks include major Big Data application areas, text analytics, graph analytics and machine-learning. Specifically our benchmark efforts focused on document categorization based on dictionary-matching, document and page ranking, and topic determination via non-negative matrix factorization. The first of the three, in particular, required innovation in benchmark creation, as there is no "golden reference" to establish correct document categorization. Existing datasets typically used as references for text-categorization assessments, such as the Enron corpus [2], are orders of magnitude smaller than what we required. Our approach for overcoming this challenge included utilizing publicly-accessible documents coded by subject, such as US Patent Office patents and applications, to create subject-specific dictionaries against which to match documents. Unique challenges of ensuring "real-world" relevance includes covering non-word terms of importance, such as band names that include regular expression characters, and a "wisdom of crowds" approach that helps us meet those challenges.

It is our intention to make our benchmark public. The benchmark is complementary to existing prescriptive benchmarks, such as Terasort and its variations that have been widely exercised in the Big Data community. In this paper, we use Terasort as a case study in Section 4.

3. WORKLOAD OPTIMIZED SYSTEMS

While industry has made substantial investments in extending its software capabilities in analytics and Big Data, thus far these new workloads are being executed on systems that were designed and configured in response to more traditional workload demands.

In this paper we present two key improvements to traditional system design. The first is the addition of reconfigurable acceleration. While reconfigurable acceleration has been used successfully in commercial systems and appliances before (e.g., DataPower® [6] and Netezza® [4]), we have demonstrated via a prototype system that such technology can benefit the processing of unstructured data.

The second innovation we discuss is a new modular and dense design that also leverages acceleration. Achieving the computational and storage densities that this design provides requires an increase in processing efficiency that is achieved by a combination of power-efficient processing cores and offloading of performance-intensive functions to the reconfigurable logic.

With an eye towards how analytics workloads are likely to evolve and executing such workloads efficiently, we conceive of a system that leverages the accelerated dense scale-out design in combination with powerful global server nodes that orchestrate and manage computation. This system can also be used to perform deeper in-memory analytics on selected data.

4. BIG DATA ON POWER SYSTEMS

Apache Hadoop [1] has been widely deployed on clusters of relatively large numbers of moderately sized, commodity servers. However, it has not been widely used on large, multi-core, heavily threaded machines even though smaller systems have increasingly large core and hardware thread counts. We describe an initial performance evaluation and tuning of Hadoop on a large multi-core cluster with only a handful of machines. The evaluation environment comprises IBM InfoSphere BigInsights [3] on a 10-machine cluster of IBM PowerLinux™ 7R2 with POWER7+ systems.

4.1 Evaluation Environment

Table 1 shows the evaluation environment, which comprises IBM InfoSphere BigInsights [3] on a 10-machine cluster of IBM PowerLinux 7R2 with POWER7+ servers. IBM InfoSphere BigInsights provides the power of Apache Hadoop in an enterprise-ready package. BigInsights enhances Hadoop by adding robust administration, workflow orchestration, provisioning and security, along with best-in-class analytics tools from IBM Research. This version of BigInsights, version 1.3, uses Hadoop 0.20.2 and its built-in HDFS file system.

Each PowerLinux 7R2 includes 16 POWER7+® cores @ 4.228GHz that can scale to 4.4GHz in a "Dynamic Power Saving - Favor Performance" mode, up to 64 hardware threads, 128 GB of memory, $6 \times 600GB$ internal SAS drives, and $24 \times 600GB$ SAS drives in an external Direct Attached Storage (DAS) drawer. We used software RAID0 over LVM for the 29 drives to each machine. One internal SAS drive is dedicated as the boot disk. The machines are connected by 10Gb Ethernet network. Each machine has two 10Gbe connections to the top of the rack switch. We used RedHat Linux (RHEL6.2). All 10 PowerLinux 7R2 with POWER7+

Table 1: Evaluation Environment

Hardware	
Cluster	10 PowerLinux 7R2 with POWER7+ Servers
CPU	16 processor cores per server (160 total)
Memory	128GB per server (1280GB total)
Internal Storage	6 600GB internal SAS drives per server (36TB total)
Storage Expansion	24 600GB SAS drives in IBM EXP24S SFF Gen2-bay Drawer, per server (144TB total)
Network	2 10Gbe connections per server
Switch	BNT BLADE RackSwitch G8264
Software	
OS	Red Hat Enterprise Linux 6.2
Java	IBM Java Version 7 SR1
BigInsights	Version 1.3 (Hadoop v0.20.2)
MapReduce	10 P7R2 with POWER7+ as DataNodes/TaskTrackers One of them as NameNode/JobTracker

machines function as Hadoop data nodes and task trackers. Note that we use one of the P7R2 with POWER7+ as both maseter node (Name Node and Job Tracker) and a slave node (Data Node and Task Tracker).

4.2 Results and Analysis

We have some early experiences with measuring and tuning standard Hadoop programs, including some of the ones used in the HiBench [11] benchmark suite, and some from real-world customer engagements. In this paper, we use Terasort, a widely used sort program included in the Hadoop distribution, as a case study. While Terasort in Hadoop can be configured to sort differing amounts of input data, we only present results for sorting 1TB of input data. In addition we compress neither input nor output data.

Our initial trial is done with the default Hadoop mapreduce configuration, e.g., limited map and reduce task slots, which does not utilize the 16 processor cores in a single PowerLinux 7R2 with POWER7+ system. As expected, the test takes hours to finish. After initial adjustment of the number of mappers and reducers to fit to the parallel processing power of 16 cores in a PowerLinux 7R2 with POWER7+ system, the execution time drastically decreases.

We then apply the following public-domain tuning methods and reference the best practices from the PowerLinux Community [8] to gradually improve the sort performance:

- Four-way Simultaneous Multithreading (SMT4) to further increase computation parallelism and stress data parallelism via large L3 caches and high memory bandwidth on POWER7+ ®;
- Aggressive read ahead setting and deadline disk IO scheduling at OS level;
- Large block size and buffer sizing in Hadoop;
- Publicly available LZO compression [7] for intermediate data compression;
- Preliminary intermediate control of map and reduce stages to better utilize available memory capacity;
- Reconfiguration of storage subsystem to remove failover support of storage adapters for effective bandwidth improvement since Hadoop handles storage failures by replication at software level;
- JVM, Jitting, and GC tuning that better fit the POWER architecture.
- Architecture featuers, like hardware prefetching, NUMA

Currently, we are able to achieve an execution time of less than 8 minutes to sort 1TB input data from disk and writing back 1TB output data to disk storage. This translates to sorting 12.8GB/node/minute for the IO intensive sort benchmark. Table 2 lists some of the BigInsights (Hadoop) job configuration parameters used in this Terasort run that completed in 7 minutes and 50 seconds.

Figure 1 shows the CPU utilization of one of the 10 nodes in the cluster during the Terasort run. All nodes have similar CPU utilization. Figure 1 shows that CPU utilization is very high during the map and map-reduce overlapping stages. As expected, it is only when all mappers finish and only reducers (380 as shown in Table 2 out of the 640 hardware threads supported by the 10 PowerLinux 7R2 with POWER7+ servers) are writing output back to disks that CPU utilization drops.

While we have large quantities of other system profile data to help us understand the potential for further performance improvement, one thing is clear: high CPU utilization does not necessarily translate into high performance. In other words, the CPU may be busy doing inefficient computing (during the Hadoop and Java framework, for instance), or performing inefficient algorithms that artificially inflate CPU utilization without improving performance. This leads us to examine full-stack optimization during the next phase of our research.

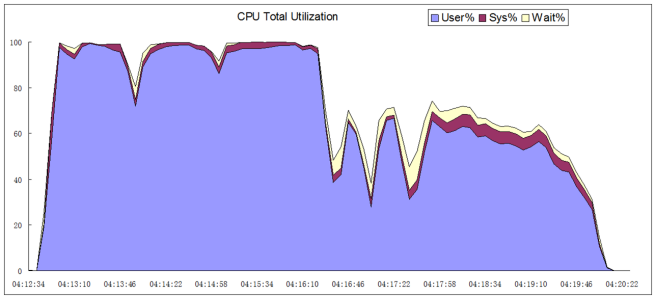

Figure 1: CPU utilization on one PowerLinux 7R2 with POWER7+ node from a Terasort run of 7 minutes 50 seconds.

As indicated above, we have only applied infrastructure tuning in this stage of the study. In the next stage, we plan to incorporate the performance enhancement features in IBM InfoSphere BigInsights for further improvement. We are also working on patches for better intermediate control in MapReduce. Newer Hadoop versions than 0.20.2 are expected to continue to deliver performance improvements.

Furthermore, we plan to apply reconfigurable acceleration technology as indicated in Section 3. In the meantime, scalability studies are also important to understand the optimal

Table 2: **Sample BigInsights (Hadoop) Job Configuration**

Parameters	Values
mapred.compress.map.output	true
mapred.map.output.compression.codec	com.hadoop.compression.lzo.LzoCodec
mapred.reduce.slowstart.completed.maps	0.01
mapred.reduce.parallel.copies	1
mapred.map.tasks	640
mapred.reduce.tasks	380
mapred.map.tasks.speculative.execution	true
io.sort.factor	120
mapred.jobtracker.taskScheduler	org.apache.hadoop.mapred.JobQueueTaskScheduler
flex.priority	0
adaptivemr.map.enable	false
io.sort.mb	650
mapred.job.reduce.input.buffer.percent	0.96
mapred.job.shuffle.merge.percent	0.96
mapred.job.shuffle.input.buffer.percent	0.7
io.file.buffer.size	524288
io.sort.record.percent	0.138
io.sort.spill.percent	1.0
mapred.child.java.opts	'-server -Xlp -Xnoclassgc -Xgcpolicy:gencon -Xms890m -Xmx890m -Xjit:optLevel=hot -Xjit:disableProfiling -Xgcthreads4 -XlockReservation'
mapred.tasktracker.map.tasks.maximum	64
mapred.tasktracker.reduce.tasks.maximum	49
dfs.replication	1
mapred.max.tracker.blacklists	20
dfs.block.size	536870912
mapred.job.reuse.jvm.num.tasks	-1

approach to (A) *strong scaling* of the BigInsights cluster in terms of scaling the number of nodes with constant input and (B) *weak scaling* of the BigInsights cluster that scales up the number of nodes with corresponding increases in input size[2].

As a preliminary proof of concept, our experiment shows that BigInsights v1.3 with only 4 PowerLinux 7R2 with POWER7+ nodes can sort 1TB input in around 20 minutes 30 seconds. Thus, the Terasort benchmark exhibits nearly linear scaling with strong scaling from 4 up through the 10 nodes in our cluster, leading to a lower system cost and footprint for situations where a 20 minutes Terasort time is acceptable.

While we have not observed that the network is a bottleneck in our 10-node cluster, this may change when the system scales up and the workload changes. Similar observations apply to our disk subsystem, which currently has 30 disks in total per PowerLinux 7R2 with POWER7+ server.

Finally, we expect people to utilize performance analysis similarly and judiciously size their systems for their particular workloads and needs. This can be useful either when they make purchasing decisions or when they reconfigure their systems in production.

5. CONCLUSIONS

In this paper, we have presented our initial study on Big Data benchmarking and methodology as well as workload optimized systems for Big Data. We have also discussed our initial experience of sorting 1TB data on a 10-node PowerLinux 7R2 with POWER7+ cluster. As of this writing, it takes less than 8 minutes to complete the sort, which translates to sorting 12.8GB/node/minute for the IO intensive sort benchmark. We expect additional improvements in the near future. We also show that 4 PowerLinux 7R2 with POWER7+ nodes can sort 1TB input with around 21 minutes. Further improvement is expected as we continue full-stack optimization on both software and hardware.

6. ACKNOWLEDGMENTS

We would like to thank the IBM InfoSphere BigInsights team for their support. Particularly, we owe a debt of gratitude to Berni Schiefer, Stewart Tate and Hui Liao for their guidance and insights. Susan Proietti Conti, Gina King, Angela S Perez, Raguraman Tumaati-Krishnan, Anitra Powell, Demetrice Browder and Chuck Bryan have been supporting us to streamline the process along the way. Last but not least, we could not have reached this stage without the generous support and advice from Dan P. Dumarot, Richard W. Bishop, Pat Buckland, Clark Anderson, Ken Blake, Geoffrey Cagle and John Williams, among others.

7. REFERENCES

[1] Apache hadoop. http://hadoop.apache.hadoop.

[2] Enron Email Dataset. http://www.cs.cmu.edu/ enron/.

[2]We borrow the two common notions of scalability, strong vs. weak scaling, from the High Performance Computing community, which we think fit well to Big Data analytics and data-intensive computing.

[3] IBM InfoSphere BigInsights. http://www-01.ibm.com/software/data/infosphere/biginsights/.
[4] IBM Netezza Data Warehouse Appliances. http://www-01.ibm.com/software/data/netezza/.
[5] IBM PowerLinux 7R2 Server. http://www-03.ibm.com/systems/power/software/linux/powerlinux/7r2/index.html.
[6] IBM WebSphere DataPower SOA Appliances. http://www-01.ibm.com/software/integration/datapower/.
[7] LZO: A Portable Lossless Data Compression Library. http://www.oberhumer.com/opensource/lzo/.
[8] PowerLinux Community. https://www.ibm.com/developerworks/group/tpl.
[9] The ClueWeb09 Dataset. http://lemurproject.org/clueweb09/.
[10] The Stanford WebBase Project. http://diglib.stanford.edu:8091/ testbed/doc2/WebBase/.
[11] S. Huang, J. Huang, J. Dai, T. Xie, and B. Huang. The HiBench benchmark suite: Characterization of the MapReduce-based data analysis. In *IEEE International Conference on Data Engineering Workshops (ICDEW)*, Long Beach, CA, USA, 2010.
[12] T. Morgan. IBM Global Technology Outlook 2012. In *Technology Innovation Exchange*, IBM Warwick, 2012.