ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology Ecology Science and Technology (IJARBEST)
Vol. I, Special Issue I, August 2015 in association with VEL TECH HIGH TECH DR. RANGARAJAN DR. SAKUNTHALA ENGINEERING COLLEGE, CHENNAI
National Conference on Recent Technologies for Sustainable Development 2015 [RECHZIG'15] - 28th August 2015

Trends in Big Data Analytics

[1]LEELAMBIKA.K.V, [2]VISHAL.G, [3]HASHIR AHAMED.R
[1]leelambika@velhightech.com, [2]vishalnishanth3@gmail.com, [3]hashir.ahamed93@gmail.com
[1]Assistant Professor, [2,3] UG Scholars
[1,2,3] Department of Computer Science and Engineering
[1,2,3]Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College

ABSTRACT

One of the major applications of future generation parallel and distributed systems is in big-data analytics.Data repositories for such applications currently exceed exabytes and are rapidly increasing in size.Beyond their sheer magnitude, these datasets and associated applications' considerations pose significantchallenges for method and software development. Datasets are often distributed and their size and privacyconsiderations warrant distributed techniques. Data often resides on platforms with widely varyingcomputational and network capabilities. Considerations of fault-tolerance, security, and access control arecritical in many applications such asDean and Ghemawat, 2004; Apache hadoop. Analysis tasks often have harddeadlines, and data quality is a major concern in yet other use applications. data-driven models and methods, capable of operating at scale, are as-yet unknown. Even when knownmethods can be scaled, validation of results is a major issue. Characteristics of hardware platforms and the software stack fundamentally impact data analytics. In this article, we provide an overview of the state of the art and focus on emerging trends to highlight the hardware, software, and application landscapeof big-data analytics. in the cloud often span multiple data centers The compute back-end of cloud made and environments typically relies on efficient and resilient data centerarchitectures built on virtualized compute and storage technologies, efficiently abstracting commodity hardware components. Current data centerstypically scale to tens of thousandsWith the development of critical Internet technologies, the visionof computing as a utility took shape in the mid 1990sThese early efforts on Grid computing typically viewed hardwareas the primary resource. Grid computing technologies focusedon sharing, selection, andaggregation of a wide variety ofgeographically distributed resources. These resources include variety of function of the system of supercomputers,storage, and other devices for solving large-scalecompute-intensive problems in science, engineering, and various commerce.transparent domain-crossing administration and resourcemanagementcapability.The concept of 'data as a resource' was popularized by peerto-peer systems Networks such as Napster, Gnutella,

andBitTorrent allowed peer nodes to share content – typically multimediadata – directly among one another in a decentralized manner.These frameworks emphasized interoperability and dynamic,ad-hoc communication and collaboration for cost reduction,resource sharing, and aggregation. However, in many of these platforms,considerations of anonymity or privacy issues and scalabilitywere secondary. Perhaps, the most visible application of big-data analytics hasbeen in business enterprises. It is estimated that a retailer fully utilizing the power of analytics can increase its operating margin by 60%. Utilizing new opportunities (for e.g., location-aware and location-based services) leads to significant potential for new revenues.A comprehensive analytics framework would require integrationof supply chain management. and many models of customer management,after-sales support advertising, etc. Business enterprises collectvast amounts of multi-modal data, including customer transactions,inventory management, store-based video feeds, advertising and customer relations, customer preferences and sentiments,sales management infrastructure, and financial data, among others.The aggregate of all data for large retailers is easily estimated to be in the exabytes, currently. With comprehensive deployment of RFIDs to track inventory, links to suppliers databases, integration with customer preferences and profiles . people in the social sciences andhave been trained to work with. Also user behavior in social networks may be correlated, which requires new methods to make sense of the causal relationships that are present in the observed stimulus–response loops. Thus, it is appropriate to embrace methodology-focused system collaboration to develop an interdisciplinary, multi-perspective approach that will yield the most interesting results for the new decision support and e-commerce issues, and other social science problems of interest. Recent hardware advances have played a major role in realizing the distributed software platforms needed for big-data analytics.Future hardware innovations — in processor technology, newer kinds of memory/storage orhierarchies, network architecture software-defined networks will continue to drive software innovations. Strong emphasis in design of these systems will be on minimizing the time spent in moving the data from storage to the processor or between storage/compute nodes in a distributed setting.

1. Introduction

With the development of critical Internet technologies, the vision of computing as a utility took shape in the mid 1990s [19].

These early efforts on Grid computing [35] typically viewed hardware as the primary resource. Grid computing technologies focused on sharing, selection, and aggregation of a wide variety of geographically distributed resources. These resources included supercomputers, storage, and other devices for solving large-scale compute-intensive problems in science, engineering, and commerce.

A key feature of these frameworks was their support fortransparent domain-crossing administration and resource management capability.

Networks such as Napster, Gnutella, anBitTorrent allowed peer nodes to share content – typically multimedia data – directly among one another in a decentralized manner.

These frameworks emphasized interoperability and dynamic,ad-hoc communication and collaboration for cost reduction, resource sharing, and aggregation. However, in many of these platforms, considerations of anonymity or privacy issues and scalability

were secondary. More recently, Cloud computing environments [95] target reliable, robust services, ubiquitously accessible (often throughbrowsers) from clients ranging from mass-produced mobile devices to general purpose computers. Cloud computing generalizes prior notions of service from infrastructure-as-a-service (computing

resources available in the cloud), and data-as-a-service (dataavailable in the cloud) to software-as-a-service (access to programs that execute in the cloud). This offers considerable benefits from points-of-view of service providers (cost reductions in hardware and administration), overall resource utilization, and better client interfaces. The compute back-end of cloud environments typically relies on efficient and resilient data center architectures, built on virtualized compute and storage technologies, efficiently abstracting commodity hardware components. Current data centers typically scale to tens of thousands of nodes and computations in the cloud often span multiple data centers.

The emerging landscape of cloud-based environments with distributed data-centers hosting large data repositories, while also providing the processing resources for analytics strongly motivates need for effective parallel/distributed algorithms. The underlying socio-economic benefits of big-data analytics and the diversity of application characteristics pose significant challenges. In the rest of this article, we highlight the scale and scope of data analytics problems.

We describe commonly used hardware platforms for executing analytics applications, and associated considerations of storage, processing, networking, and energy. We then focus on the software substrates for applications, namely virtualization technologies, runtime systems/execution environments, and programming models. We conclude with a brief discussion of the diverse applications of data analytics, ranging from health and human welfare

to computational modeling and simulation.

1.1. Scale and scope of data analytics

Recent conservative studies estimate that enterprise server systems in the world have processed 9.57×1021 bytes of data in 2008

[83]. This number is expected to have doubled every two years from that point. As an example, Walmart servers handle more than one million customer transactions every hour, and this information is inserted into databases that store more than 2.5 petabytes of data—the equivalent of 167 times the

number of books in the Library of Congress [94]. The Large Hadron Collider at CERN will produce roughly 15 petabytes of data annually—enough to fill more than 1.7 million dual-layer DVDs per year [60]. Each day, Facebook operates on nearly 500 terabytes of user log data and several hundreds of terabytes of image data. Every minute, 100 h of video are uploaded on to YouTube and upwards of 135,000 h are watched [98]. Over 28,000 multi-media (MMS) messages are sent every second [3]. Roughly 46 million mobile apps were downloaded in 2012, each app collecting more data. Twitter [87] serves more than 550 million active users, who produce 9100 tweets every second. eBay systems process more than 100 petabytes of data every day [64]. In other domains, Boeing jet engines can produce 10 terabytes of operational information for every 30 min of operation.

This corresponds to a few hundred terabytes of data for a single Atlantic crossing, which, if multiplied by the 25,000 flights each day, highlights the data footprint of sensor and machine-produced nformation.

These examples provide a small glimpse into the rapidly expanding ecosystem of diverse sources of massive datasets currently in existence. Data can be structured (e.g., financial, electronic medical records, government statistics), semi-structured (e.g., text, tweets, emails), unstructured (e.g., audio and video), and real-time (e.g., network traces, generic monitoring logs). All of these applications share the potential for providing invaluable insights, if organized and analyzed appropriately.

Applications requiring effective analyses of large datasets are widely recognized today. Such applications include health care analytics (e.g., personalized genomics), business process optimization, and social-network-based recommendations. However, projections suggest that data growth will largely outpace foreseeable improvements in the cost and density of storage technologies, the available computational power for processing it, and the associated energy footprint. For example, between 2002 and 2009 data traffic grew 56-fold, compared to a corresponding 16-fold increase in computing power (largely tracking Moore's law). In comparison, between 1998 and 2005 data centers grew in size by 173% per year [68]. Extrapolating these trends, it will take about 13 years for a 1000-fold increase in computational power (or theoretically 1000× more energy). However, energy efficiency is not expected to increase by a factor of over 25 over the same time period. This generates a severe mismatch of almost a 40-fold increase in the data analytics energy footprint.

Workload characteristics. A comprehensive study of big-data workloads can help understand their implications on hardware and software design. Inspired by the seven dwarfs of numerical computation [11], Mehul Shah et al. [82] attempt to define a set of ''data dwarfs''—meaning key data processing kernels—that provide current and future coverage of data-centric workloads. Drawing from an extensive set of workloads, they establish a set of classifying dimensions (response time, access pattern, working set, data type, read vs write, processing complexity) and conclude that five workload models could satisfactorily cover data-centric workloads as of 2010: (i) distributed sort at petabytes scale, (ii) in-memory index search, (iii) recommendation system, featuring high processing load and regular communication patterns, (iv) sequential-access based data de-duplication and (v) video uploading and streaming server at interactive response rates. While Online Analytic Processing (OLAP) workloads can readily be expressed as a combination of (i), (iii) and (iv), Online Transaction Processing (OLTP) workloads can only be partially captured and might require another category in the future; in-memory index and query support captures some facets of these workloads, but the working sets can become too large to fit in memory.

1.2. Design considerations

The scale, scope and nature (workload characteristics) of big data analytics applications, individually, provide interesting insights into the design and architecture of future hardware and software systems.

Impact on hardware. The data access patterns and more specifically the frequency of how data is accessed (cold versus hot data) can drive future memory hierarchy optimizations: data generally starts being hot; however as time progresses, it becomes archival, cold, most suitable for storage in NVMs. However, there are notable exceptions of periodicity or churn in access patterns (season-related topics, celebrity headlines) and concurrently hot massive datasets (comparative genomic calculations) that should be taken into consideration.

Furthermore, latent correlations among dimensions can arise with hardware stack projections: a single video, due to multiple formats or language subtitles, results in many versions. These could either be generated offline and stored (thus needing ample storage) or generated on the fly (transcoding and translation on demand) putting pressure on the computing infrastructure of the data-center, or alternatively on the user's device (client-side computing).

Alternatively, one might have to rethink the relative prioritization of advances in processor designs over the performance of the I/O subsystem—a common assumption in current architecture design. At the extreme of such an alternative, an option would be the consideration of a possible ''inversion'': a hierarchy of compute elements supporting the data store instead of today's designs of memory hierarchies to serve the compute element. Gradually collapsing existing storage hierarchies would smoothen such a transition and further provide savings in energy consumption.

Understanding the workloads could also identify opportunities for implementing special purpose processing elements directly K. Kambatla et al. / J. Parallel Distrib. Comput. 74 (2014) 2561–2573 2563 in hardware. GPUs, field-programmable gate arrays (FPGAs), specialized application-specific integrated circuits (ASICs), and dedicated video encoders/decoders warrant consideration. Such hardware accelerators drastically reduce the energy consumption, compared to their general-purpose processing counterparts. These could be integrated on-chip, leading to families of data-centric asymmetric multiprocessors [66].

NVMs offer latency reduction of approximately three orders of magnitude (microseconds) over this time. There are proposals for using flash-based solid-state-disks (SSDs) to support key-value store abstractions, for workloads that favor it. Yet others propose to organize SSDs as caches for conventional disks (hybrid design).

Ideally the persistence of NVMs should be exposed at the instruction-set level (ISA), so that the operating systems can utilize them efficiently (e.g., by redesigning parts that assume memory volatility or provide, to the upper layers, an API for placing archival data on energy-efficient NVM modules). On the other hand, the capability of durable memory writes reduces isolation; this issue could be addressed via durable memory transactions [93,24]. From the perspective of algorithm design and related data structures, non-volatility could push towards alternate, optimized designs and implementations of index structures, [22], key-value stores [91], database and file systems [27,12], all integral components of bigdata analytics.

2.2. Processing landscape for data analytics

Chip multiprocessors (CMPs) are expected to be the computational work-horses for big data analytics. It seems however that there is no consensus on the specifics of the core ensemble hosted on a chip. Basically there are two dimensions of differentiation, operations with suitably engineered, virtualization friendly equivalents. However, altering the source code of an operating system can be problematic due to licensing issues, and it potentially introduces incompatibilities. In an alternate approach, a binary translator runs the non-virtualizable, privileged parts and dynamically patches the ‘‘offending’’ instructions, also retaining in a trace cache the translated blocks for optimization purposes.

For memory management, VMM maintains a shadow of each virtual machine's memory-management data structure, its shadow page table. VMM updates these structures reflecting operating system's changes and establishes the mapping to actual pages in the hardware memory. Challenges here include enabling the VMM to leverage the operating system's internal state for efficient paging in/out and sharing identical physical pages across multiple virtual machines monitored by a single VMM. This sharing will be particularly important for homogeneous pools (in terms of software configuration) of virtual machines executing, over multicore multiprocessors on-a-chip, the workloads of big data analytics in the future.

References

[1] Daniel J. Abadi, Don Carney, Ugur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, Stan Zdonik, Aurora: a new model and architecture for data stream management, VLDB J. 12 (2) (2003) 120–139.

[2] Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel Abadi, Alexander Rasin, Avi Silberschatz, HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads, in: VLDB, 2009.

[3] http://agbeat.com/tech-news/how-carriers-gather-track-and-sell-yourprivate-data/.

[4] Yanif Ahmad, Bradley Berg, U˘gur Cetintemel, Mark Humphrey, Jeong-Hyon Hwang, Anjali Jhingran, Anurag Maskey, Olga Papaemmanouil, Alexander Rasin, Nesime Tatbul, Wenjuan Xing, Ying Xing, Stan Zdonik, Distributed operation in the borealis stream processing engine, in: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD'05, ACM, New York, NY, USA, 2005, pp. 882–884.

[5] Mohammad Al-Fares, Alexander Loukissas, Amin Vahdat, A scalable, commodity data center network architecture, in: Victor Bahl, David Wetherall, Stefan Savage, Ion Stoica (Eds.), SIGCOMM, ACM, 2008, pp. 63–74.

[6] Mohammad Alizadeh, Albert G. Greenberg, David A. Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, Murari Sridharan, Data center TCP (DCTCP), in: Shivkumar Kalyanaraman, Venkata N. Padmanabhan, K.K. Ramakrishnan, Rajeev Shorey, Geoffrey M. Voelker (Eds.), SIGCOMM, ACM, 2010, pp. 63–74.

[7] David G. Andersen, Jason Franklin, Michael Kaminsky, Amar Phanishayee, Lawrence Tan, Vijay Vasudevan, FAWN: a fast array of wimpy nodes, Commun. ACM 54 (7) (2011) 101–109.

[8] Rasmus Andersen, Brian Vinter, The scientific byte code virtual machine, in: GCA, 2008, pp. 175–181.

[9] H. Andrade, B. Gedik, K.L. Wu, P.S. Yu, Processing high data rate streams in system S, J. Parallel Distrib. Comput. 71 (2) (2011) 145–156.

[10] Luiz André Barroso, Urs Hölzle, The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, in: Synthesis Lectures on Computer Architecture, Morgan & Claypool Publishers, 2009.

[11] Krste Asanovic, Ras Bodik, Bryan Christopher Catanzaro, Joseph James Gebis,

Parry Husbands, Kurt Keutzer, David A. Patterson, William Lester Plishker, John Shalf, Samuel Webb Williams, Katherine A. Yelick, The Landscape of Parallel Computing Research: A View from Berkeley. Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, 2006.

[12] M. Athanassoulis, A. Ailamaki, S. Chen, P. Gibbons, R. Stoica, Flash in a DBMS: where and how? Bull. IEEE Comput. Soc. Tech. Committee Data Eng. (2010).

[13] Jason Baker, Chris Bond, James C. Corbett, J.J. Furman, Andrey Khorlin, James Larson, Jean-Michel Leon, Yawei Li, Alexander Lloyd, Vadim Yushprakh, Megastore: Providing Scalable, Highly Available Storage for Interactive Services, in: CIDR'11, 2011.

[14] J. Baliga, R.W.A. Ayre, K. Hinton, R.S. Tucker, Green cloud computing: balancing energy in processing, storage, and transport, Proc. IEEE 99 (1) (2011) 149–167.

[15] Costas Bekas, Alessandro Curioni, A new energy aware performance metric, Comput. Sci.-Res. Dev. 25 (3–4) (2010) 187–195.

[16] K. Birman, D. Freedman, Qi Huang, P. Dowell, Overcoming cap with consistent soft-state replication, Computer 45 (2) (2012) 50–58.

[17] E.A. Brewer, Towards robust distributed systems, in: Proc. 19th Annual ACM Symposium on Priniciples of Distributed Computing, PODC, 2000, pp. 7–10.

[18] Mike Burrows, The chubby lock service for loosely-coupled distributed systems, in: Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI'06, USENIX Association, Berkeley, CA, USA, 2006, pp. 335–350.