Front. Cur. Trends. Engg. Tech. SI:1, pp. 38 – 42 (2016)
OPEN ACCESS **ISSN:** ***2456 – 1185*** Available online: www.fctet.in

An International Open Access Journal
Published by ACET

Short Note: ELECTRIAL, ELECTRONICS & COMPUTER SCIENCE

Scalable many-core architecture for Big Data towards cost minimization

M. Vigilson Prem[1]and M. Jeevabharathi[2]

Abstract - In the field of data processing, the process of examining large amounts of different types of data, or Big-Data, in an effort to uncover hidden patterns or unknown correlations has become a major need in our society. In this context, stream mining applications are now widely used in several domains such as financial analysis, video annotation, surveillance, medical services, traffic prediction, etc. In order to cope with the Big-Data stream input and its highvariability, modern stream mining applications implement systems with heterogeneous classifiersand adapt online to its input data stream characteristics variation. Moreover, estimating the energyconsumption of applications is one of the key aspects in optimizing embedded systems energyconsumption with the higher data transmission. The total energy includes the energy consumption ofthe processor core, flash memory, memory controller, and static random- access memory. The modelparameters are instructions opcode, number of shift operations, register bank bit flips, instructionsweight and their hamming distance, and different types of memory accesses. Furthermore, the effectof pipeline stalls has been considered. In order to validate the proposed stream mining model, aphysical hardware platform equipped with energy measurement capabilities was developed.Therefore, in this work propose a novel low-power many core architecture for stream miningapplications that is able to cope with the dynamic data-driven nature of stream mining applicationswhile consuming limited power. These works exploration indicates that thisnew proposed architecture is able to adapt to different classifiers for its multiple scalable vector processing units andtheir re-configurability feature at runtime. Moreover, this platform architecture includes a memoryhierarchy optimized for Big-Data streaming and implements modern fine-grained power managementtechniques over all the different types of cores allowing then minimum energy consumption for eachkind of executed classifier. Furthermore, this work proposes a simple yet accurate instruction level energy estimation model for embedded systems along with the higher data.

[1]Department of Computer Science Engineering, RMD Engineering College, Chennai, Tamil Nadu

[2]Department of Electronics and CommunicationEngineering, Sri Shanmugha college of Engineering and Technology, Sankari, Salem, Tamil Nadu

Keywords: *big data, dataflow, data placement, distributed datacenters, cost minimization, task assignment.*

1. INTRODUCTION

Data explosion in recent years leads to a rising demand for big data processing in modern data centers that are usually distributed at different geographic regions many efforts have been made to lower the computation or communication cost of data centers. Data center resizing (DCR) has been proposed to reduce the computation cost by adjusting the number of activated servers via task placement. Based on DCR, some studies have explored the geographical distribution nature of data centers and electricity price heterogeneity to lower the electricity cost. Big data service frameworks, e.g.,, comprise a distributed file system underneath, which distributes data chunks and their replicas across the data centers for fine-grained load-balancing and high parallel data access performance. To reduce the communication cost, a few recent studies make efforts to improve data locality by placing jobs on the servers where the input data reside to avoid remote data loading.

Although the above solutions have obtained some positive results, they are far from achieving the cost efficient big data processing because of the following weaknesses. First, data locality may result in a waste of resources. For example, most computation resource of a server with less popular data may stay idle. TheLow resource utility further causes more servers to be activated and hence higher operating cost.

Second, the links in networks vary on the transmission rates and costs according to their unique features, e.g., the distances and physical optical fiber facilities between data centers. However, the existing routing strategy among data centers fails to exploit the link diversity of data center networks. Due to the storage and computation capacity constraints, not all tasks can be placed onto the same server, on which their corresponding data reside. It is unavoidable that certain data must be downloaded from a remote server. In this case, routing strategy matters on the transmission cost. As indicated by Jin et al., the transmission cost, e.g., energy, nearly proportional to the number of network link used. The more link used, the higher cost will be incurred. Therefore, it is essential to lower the number of links used while satisfying all the transmission requirements.

Third, the Quality-of-Service (QoS) of big data tasks has not been considered in existing work. Similar to conventional cloud services, big data applications also exhibit Service-Level-Agreement (SLA) between a service provider and the requesters. To observe SLA, a certain level of QoS, usually in terms of task completion time, shall be guaranteed. The QoS of any cloud computing tasks is first determined by where they are placed and how many computation resources are allocated. Besides, the transmission rate is another influential factor since big data tasks are data-centric and the computation task cannot proceed until the corresponding data are available. Existing studies, e.g., on general cloud computing tasks mainly focus on the computation capacity constraints, while ignoring the constraints of transmission rate.

To conquer above weaknesses, we study the cost minimization problem for big data processing via joint optimization of task assignment, data placement, and routing in geo-distributed data centers. Specifically, we consider the following issues in our joint optimization. Servers are equipped with limited storage and computation resources. Each data chunk has a storage requirement and will be required by big data tasks. The data placement and task assignment are transparent to the data users with guaranteed QoS.

Our objective is to optimize the big data placement, task assignment, routing and DCR such that the overall computation and communication cost is minimized. To describe the rate-constrained computation and transmission in big data processing process, we propose a two dimensional Markov chain and derive the expected task completion time in closed form. To deal with the high computational complexity of solving MINLP, we linarite it as a mixed-integer linear programming (MILP) problem, which can besolved using commercial solver. Through extensive numerical studies, we show the high efficiency of our proposed joint-optimization based algorithm.

2. RELATED WORKS

2.1 Server Cost Minimization

Large-scale data centers have been deployed all over the world providing services to hundreds of thousands of users. According to [11], a data center may consist of large numbers of servers and consume megawatts of power. Millions of dollars on electricity cost have poseda heavy burden on the operating cost to data centerproviders. Therefore, reducing the electricity cost has received significant attention from both academia and industry [5], [11]–[13]. Among the mechanisms that have been proposed so far for data center energy management, the techniques that attract lots of attention are task placement and DCR. DCR and task placement are usually jointly considered to match the computing requirement. Liu et al. [4] re-examine the same problem by taking network delay into consideration. Fan et al. [12] study power provisioning strategies on how much computing equipment can be safely and efficiently hosted within a given power budget.Rao et al. [3] investigate how to reduce electricitycost by routing user requests to geo-distributed datacenters with accordingly updated sizes that match the requests. Recently, Gao et al. [14] propose the optimal workload control and balancing by taking account of latency, energy consumption and electricity prices. Liuet al. [15] reduce electricity cost and environmental impact using a holistic approach of workload balancing that integrates renewable supply, dynamic pricing, andcooling supply.

2.2. Data Placement

Shachnai et al. [21] investigate how to determine a placement of Video-on-Demand (VoD) file copies on the servers and the amount of load capacity assigned to each file copy so as to minimize the communication cost while ensuring the user experience. Agarwal et al. [22] propose an automated data placement mechanism Volley for geo-distributed cloud services with the consideration of WAN bandwidth cost, data center capacity limits, data inter-dependencies, etc. Cloud services make use of Volley by submitting logs of datacenter requests. Volley analyzes the logs using an iterative optimization algorithm based on data access patterns and client locations, and outputs migration recommendations back to the cloud service. Cidon et al. [23] invent Min copy sets, a data replication placement scheme that decouples data distribution and replication to improve the data durability properties in distributed data centers. Recently, Jin et al. [10] propose a joint optimization scheme that simultaneously optimizes virtual machine (VM) placement and network flow routing to maximize energy savings. Existing work on data center cost optimization, big data management or data placement mainly focuses on one or two factors. To deal with big data processing in geo-distributed data centers, we argue that it is essential to jointly consider data placement, task assignment and data flow routing in a systematically way.

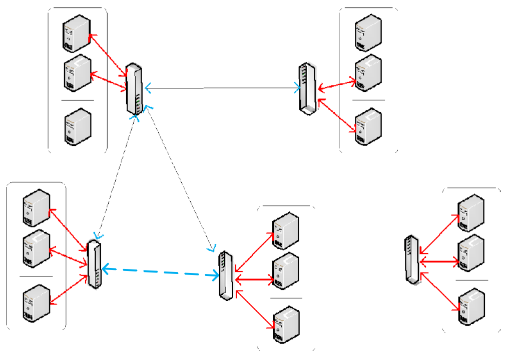

Fig.1. Data center topology

3. TASK MODEL

We consider a geo-distributed data center topology as shown in Fig. 1, in which all servers of the same data center (DC) are connected to their localswitch, while data centers are connected through switches. There are a set I of data centers, and each data center i ∈ I consists of a set J of servers that are connected to a switch mii∈ M with a local transmission cost of CL . In general, the transmission cost C for inter-data center traffic is greater than CL, i.e., CRR. Withoutloss of generality, all servers in the network have the same computation resource and storage capacity, both ofwhich are normalized to one unit. We use J to denotethe set of all severs,i.e., J = J1∪J2>C· · ·∪J|I|L.The whole system can be modeled as a directed graphG = (N, E). The vertex set N = M∪J includes theset M of all switches and the set J of all servers, andE is the directional edge set. All servers areconnectedto, and only to, their local switch via intra-data centerlinks while the switches are connected via inter-datacenter links determined by their physical connection.The weight of each link w, representing the correspondingcommunication cost, can be defined as(u;v)

w(u;v)={CR, if u, v ∈ M,CL, otherwise.(1)

3.1 Data Uploading

Select the big data and stored into the hadoop environment for the performing map reduce on hadoop. The data should be loaded into the VM server location. After uploading the file the data segmentation is performed for further process.

3.1.1. Segmentation

Packet segmentation improves network performance by splitting the packets in received Ethernet frames into separate buffers. Packet segmentation may be responsible for splitting one into multiple so that reliable transmission of each one can be performed individually. Segmentation may be required when the data packet is larger than the maximum transmission unit supported by the network.

The packet processing system is specifically designed for dealing with the network traffic. Most networks, such as the Internet, are distributed and layered systems composed of hosts, workstations, switches and routers etc.The processing speed of edge equipment falls behind those in core network. Finally the access network connect s the terminals of a customer endpoint. And usually the bandwidth and line rate requirement is lowest among the three.

The packet processing system can be equipped in any layer of the network, either in the high end core routers or in the LAN switches. The flexibility of the system comes from the programmable elements within it, i.e. NPs. And a series of stacked network protocols guarantee its capability to achieve the performance specification.

3.2. Task Assignment

The Data Center should be selected according to computation and storage capacity of servers resides in the data center. Identification of Data Center is important matter for minimizing operational expenditure of servers reside in the each data centers. Data chunks can be placed in the same data center when more servers are provided in each data center.Further increasing the number of servers will not affect the distributions of tasks. Task should be assigned to data center where number of activated servers is optimal. Task assignment is deeply influence the operational expenditure of data center. Task is assigned to data center according to nearest data center for effectively processing of data. Each data chunk has a storage requirement and will be required by big data tasks.

3.3. Data Loading

A Data Placement on the servers and the amount of load capacity assigned to each file copy so as to minimize the communication cost while ensuring the user experience. Cloud services make use of Volley by submitting logs of datacenter requests. Volley analyzes the logs using an iterative optimization algorithm based on data access patterns and client locations, and outputs migration recommendations back to the cloud service. Invent Min Copy sets, a data replication placement scheme that decouples data distribution and replication to improve the data durability properties in distributed data centers. Recently, Jin et propose a joint optimization scheme that simultaneously optimizes virtual machine (VM) placement and network flow routing to maximize energy savings.

3.4. Processing of Task

The high computational server should not process the low population of data chunk. Because it increases the operational expenditure of server, wastage of storage and transmission cost. The population of data is processed depend upon the computational capacity of servers reside in the data centers.

3.5. Evaluation Process

We present the performance results of our joint-optimization algorithm using the MILP formulation. Evaluate server cost, communication cost and overall cost under different total server numbers.

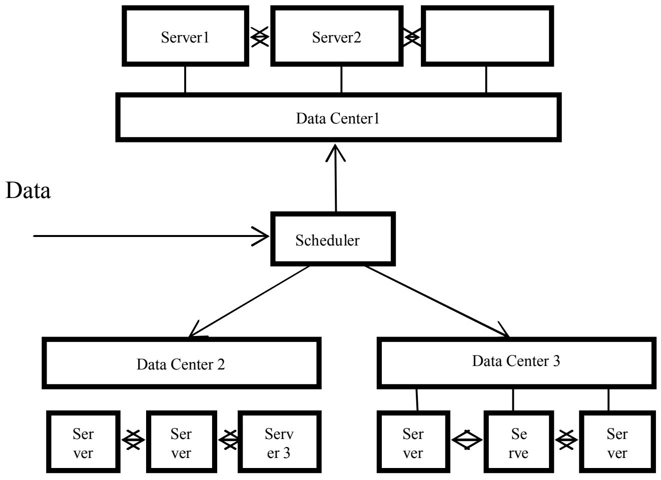

Fig 2. System architecture

4. BACKGROUNDS AND RELATED WORK

The MapReduce framework was first advocated by Google in 2004 as a programming model for its internal massive data processing [33]. Since then it has been widely discussed and accepted as the most popular paradigm for data intensive processing in different contexts. Therefore there are many implementations of this framework in both industry and academia (such as Hadoop [34], Dryad [35], Greenplum [36]), each with its own strengths and weaknesses. Since Hadoop MapReduce is the most popular open source implementation, it has become the *de facto* research prototype on which many studies are conducted. We thus use the terminology of the Hadoop community in the rest of this paper, and focus here mostly on related work built using the Hadoop implementation.

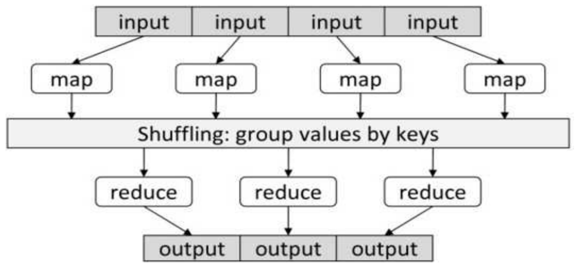

Fig. 3: MapReduce framework.

From an abstract viewpoint, a Map Reduce job essentially consists of two sets of tasks: map tasks and reduce tasks, as shown in Fig. 3.

The executions of both sets of tasks are synchronized into a map stage followed by a reduce stage. In the map stage, the entire dataset is partitioned into several smaller chunks in forms of key value pairs, each chunk being assigned to a map node for partial computation results. The map stage ends up with a set of intermediate key-value pairs on each map node, which are further shuffled based on the intermediate keys into a set of scheduled reduce nodes where the received pairs are aggregated to obtain the final results. For an iterative Map Reduce job, the final results could be tentative and further partitioned into a new set of map nodes for the next round of the computation. A batch of Map Reduce jobs may have multiple stages of MapReduce computation, each stage running either map or reduce tasks in parallel, with enforced synchronization only between them. Therefore, the executions of the jobs can be viewed as a fork&join workflow characterized by multiple synchronized stages, each consisting of a collection of sequential or parallel map/reduce tasks. An example of such a work flow is shown in Fig. 3 which is composed of 4 stages, respectively with 8, 2, 4 and 1 (map or reduce) tasks. These tasks are to be scheduled on different nodes for parallel execution. However, in heterogeneous clouds, different nodes may have different performance and/or configuration specifications, and thus may have different service rates. Therefore, because resources are provisioned on-demand in cloud computing, the CSPs are faced with a general

Practical problem: how are resources to be selected and utilized for each running task in a cost-effective way? This problem is, in particular, directly relevant to CSPs wanting to compute their MapReduce workloads, especiallywhen the computation budget is fixed.Hadoop MapReduce is made up of an executionruntime and a distributed file system. The executionruntime is responsible for job scheduling and execution.It is composed of one master node called *JobTracker* andmultiple slave nodes called *TaskTrackers*. The distributedfile system, referred to as *HDFS*, is used to manage taskand data across nodes. When the JobTracker receives a submitted job, it first splits the job into a number of map and reduce tasks and then allocates them to the Task Trackers, as described earlier. As with most distributed systems, the performance of the task scheduler greatly affects the scheduling length of the job, as well as, in our particular case, the budget consumed. Hadoop MapReduce provides a FIFO-based default scheduler at job level, while at task level; it offers developers a Task Scheduler interface to design their own schedulers. By default, each job will use the whole cluster and execute in order of submission. In order to overcome this inadequate strategy and share fairly the cluster among jobs and users over time, Facebook and Yahoo! leveraged the interface to implement Fair Scheduler and Capacity Scheduler, respectively. Beyond fairness, there exists additional research on the Scheduler of Hadoop Map Reduce aiming at improving its scheduling policies.

5. CONCLUSION

In this paper, we jointly study the data placement, task assignment, data center resizing and routing to minimize the overall operational cost in large-scale geo-distributed data centers for big data applications. We first characterize the data processing process using a two-dimensional Markov chain and derive the expected completion time in closed-form, based on which the joint optimization is formulated as an MINLP problem. To tackle the high computational complexity of solving our MINLP, we linearize it into an MILP problem. Through extensive experiments, we show that our joint-optimization solution has substantial advantage over the approach by two-step separate optimization. Several interesting phenomena are also observed from the experimental results.

REFERENCES

[1]"DataCenterlocations,"http://www.google.com/about/datacenters /inside/locations/index.html.

[2]R.Raghavendra, P.Ranganathan, V.Talwar, Z.Wang, and X.Zhu, "No"Power"Struggles:CoordinatedMulti-levelPower Manage-ment forthe Data Center,"inProceedingsofthe13thInternationalConferenceonArchitecturalSupportforProgrammingLanguagesand OperatingSystems (ASPLOS). ACM,2008,pp. 48–59.

[3] L.Rao,X. Liu,L.Xie,and W.Liu,"MinimizingElectricity Cost: Optimizationof DistributedInternetData Centersin a Multi-Electricity-MarketEnvironment,"inProceedingsofthe 29thInterna-tionalConferenceonComputerCommunications(INFOCOM). IEEE,2010,pp. 1–9.

[4] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L.Andrew, "GreeningGeographicalLoad Balancing,"in Proceedings ofIn-ternationalConferenceon Measurement andModeling ofComputer Systems (SIGMETRICS). ACM,2011,pp. 233–244.

[5] R.Urgaonkar,B.Urgaonkar,M. J.Neely, and A. Sivasubrama-niam, "OptimalPower Cost ManagementUsing Stored Energy in Data Centers," in Proceedingsof International Conferenceon Measurement and Modeling ofComputer Systems (SIGMETRICS). ACM,2011,pp. 221–232.

[6] B. L.Hong Xu,Chen Feng,"TemperatureAware Workload Managementin Geo-distributed Datacenters,"in Proceedingsof InternationalConferenceonMeasurementandModelingofComputer Systems (SIGMETRICS). ACM,2013,pp. 33–36.

[7] J. Dean and S.Ghemawat,"Mapreduce:simplifieddata process-ingonlarge clusters,"CommunicationsoftheACM, vol.51,no.1, pp. 107–113,2008.

[8] S.A. Yazd, S.Venkatesan,and N. Mittal, "Boosting energy ef-ficiency with mirroreddata block replicationpolicy and energy scheduler,"SIGOPSOper.Syst.Rev.,vol.47, no.2,pp.33–40,2013. [9] I.Marshalland C.Roadknight,"Linkingcache performanceto user behaviour,"Computer Networks andISDN Systems, vol. 30, no.223,pp. 2123–2130,1998.

[10] H. Jin,T.Cheocherngngarn,D. Levy, A. Smith, D. Pan, J.Liu, and N.Pissinou,"Joint Host-NetworkOptimizationforEnergy-Efficient Data Center Networking," in Proceedingsof the 27th InternationalSymposium onParallelDistributedProcessing(IPDPS),2013,pp. 623–634.

[11] A.Qureshi,R.Weber, H.Balakrishnan,J. Guttag,and B. Maggs, "Cuttingthe Electric BillforInternet-scaleSystems,"inProceed- ings of the ACM SpecialInterest Group on Data Communication (SIGCOMM). ACM,2009,pp. 123–134.

[12] X.Fan,W.-D.Weber, and L.A.Barroso, "PowerProvisioningfor AWarehouse-sizedComputer,"inProceedingsofthe34thAnnual InternationalSymposium onComputerArchitecture(ISCA). ACM,2007,pp. 13–23.

[13] S.Govindan,A.Sivasubramaniam,and B.Urgaonkar, "Benefits and LimitationsofTappingInto Stored Energy forDatacenters,"

inProceedingsofthe38thAnnual InternationalSymposium onComputerArchitecture(ISCA). ACM,2011,pp. 341–352.

[14] Gao, A.R.Curtis, B.Wong, and S.Keshav, "It'sNot Easy BeingGreen," inProceedingsoftheACM SpecialInterestGroupon DataCommunication(SIGCOMM). ACM,2012,pp. 211–222.

[15] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang,M.Marwah,andC.Hyser, "RenewableandCooling Aware Work-load ManagementforSustainableDataCenters,"inProceedingsof InternationalConferenceonMeasurementandModelingofComputer Systems (SIGMETRICS). ACM,2012,pp. 175–186.

[16] M.Sathiamoorthy,M.Asteris, D.Papailiopoulos,A.G.Dimakis,R.Vadali, S. Chen, and D.Borthakur,"Xoring elephants:novel erasurecodes forbigdata,"inProceedingsofthe39thinternational conferenceon Very LargeData Bases,ser. PVLDB'13. VLDB Endowment,2013,pp. 325–336.

[17] B. Hu, N.Carvalho,L.Laera, and T.Matsutsuka,"Towardsbig linked data: a large-scale,distributed semantic data storage," in Proceedingsofthe 14th International Conferenceon Information IntegrationandWeb-based Applications&Services,ser. IIWAS'12. ACM,2012,pp. 167–176.

[18] J. Cohen, B. Dolan, M.Dunlap,J. M.Hellerstein,and C.Welton, "Mad skills: new analysispracticesfor big data,"Proc.VLDB Endow.,vol.2,no.2,pp. 1481–1492,2009.

[19] R.Kaushikand K.Nahrstedt,"T*: Adata-centriccooling energy costsreductionapproachforBigDataanalyticscloud," in2012InternationalConferenceforHighPerformanceComputing,Networking, StorageandAnalysis(SC),2012,pp. 1–11.

[20] F.Chen, M.Kodialam,and T.V.Lakshman,"Joint schedulingofprocessingand shuffle phasesinmapreducesystems,"inProceedingsofthe29thInternationalConferenceonComputerCommunications (INFOCOM). IEEE,2012,pp. 1143–1151.

[21] H. Shachnai,G. Tamir, and T.Tamir, "Minimalcost reconfiguration ofdata placementin astoragearea network,"Theoretical ComputerScience,vol.460,pp. 42–53,2012.

[22] S.Agarwal,J. Dunagan,N.Jain,S.Saroiu, A.Wolman,and H.Bhogan, "Volley: AutomatedData Placementfor Geo-Distributed Cloud Services," in The 7th USENIX Symposium on Networked Systems DesignandImplementation(NSDI), 2010,pp. 17–32.

[23] A. Cidon, R.Stutsman,S.Rumble, S.Katti, J.Ousterhout, andM.Rosenblum,"MinCopysets:DerandomizingReplication In Cloud Storage," in The 10th USENIXSymposium on Networked Systems DesignandImplementation(NSDI), 2013.