Large Scale Data Processing from Multiple Data Centers

S. Ramya[1], T.V.K.P. Prasad[2],
PG scholar[1], Assistant professor[2],
Department of Computer science and Engineering[1, 2]
SRKR Engineering College, Bhimavaram[1,2].

Abstract- With the globalization of service, organizations continuously produce large volumes of data that need to be analyzed over geo-dispersed locations. Traditionally central approach that moving all data to a single cluster is inefficient or infeasible due to the limitations such as the scarcity of wide-area bandwidth and the low latency requirement of data processing. Processing big data across geo-distributed datacenters continues to gain popularity in recent years. However, managing distributed Map Reduce computations across geo-distributed datacenters poses a number of technical challenges: how to allocate data among a selection of geo-distributed datacenters to reduce the communication cost, how to determine the Virtual Machine provisioning strategy that offers high performance and low cost, and what criteria should be used to select a datacenter as the final reducer for big data analytics jobs. In this paper, these challenges is addressed by balancing bandwidth cost, storage cost, computing cost, migration cost, and latency cost, between the two Map Reduce phases across datacenters. We formulate this complex cost optimization problem for data movement, resource provisioning and reducer selection into a joint stochastic integer nonlinear optimization problem by minimizing the five cost factors simultaneously. An efficient online algorithm that is able to minimize the long-term time-averaged operation cost is further designed.

Keywords: Big Data Processing; Cloud Computing; Data Movement; Virtual Machine Scheduling; Online Algorithm

I. Introduction

We are entering a big data era with more data generated and collected in a geographically distributed manner in many areas such as finance, medicine, social web, astronomy etc. With the increasing explosion of distributed data, the huge treasures hidden in it are waiting for us to explore for providing valuable insights. To illustrate, social web sites such as Face book can uncover usage patterns and hidden correlations by analyzing the web site history records (e.g., click records, activity records et al.) to detect social hot event and facilitate its marketing decision (e.g., advertisement recommendation), and the Square Kilometer Array (SKA), an international project to build the world's largest telescope distributed over several countries, need to fusion the geographically dispersed data for scientific applications. However, due to the properties such as large-scale volume, high complexity and depressiveness of big data coupled with the scarcity of Wide area bandwidth (e.g., trans-oceanic link), it is inefficient and/or infeasible to process the data with centralized solutions.

This has fueled strong companies from industry to deploy multi datacenter cloud and hybrid cloud. These cloud technologies offer a powerful and cost-effective solution to deal with increasingly high velocity of big data generated from geo-distributed sources (e.g., Face book, Google and Microsoft etc). For majority of the common organizations (e.g., SKA), it is economic to rent resource from public cloud, with considering the advantages of cloud computing such as flexibility and pay-as-you-go business cloud computing such as flexibility and pay-as-you-go business model. Map Reduce is a distributed programming model for processing large-scale dataset in parallel, which has shown its outstanding effectiveness in many existing applications . Since original Map Reduce model is not optimized for deployment across datacenters, aggregating distributed data to a single datacenter for centralized processing is a widely-used approach.

However, waiting for such centralized aggregation suffers from significantly delays due to the heterogeneous and limited bandwidth of user cloud link. Notice that the bandwidth of inter-datacenter link is usually dedicated relatively high-bandwidth lines, moving the data to multiple datacenters for map operation in parallel and then aggregating the intermediate data to a single datacenter for reduce operation using inter-datacenter link has potential to reduce the latency. Furthermore, different kinds of cost (e.g., incurred by moving data or renting VM) also can be optimized considering the heterogeneity of the link speed, the dynamism of the data generation and the resource price. Therefore, distributing data from multi-sources into multi-datacenters and processing them using distributed Map Reduce is an idea way to deal with the large volume dispersed data. Hitherto, the most important questions to be solved include:

1) how to optimize the placement of large-scale datasets from various locations onto geo-distributed datacenter cloud for processing and
2) how many resources such as computing resources should be provisioned to guarantee performance and availability while minimizing the cost. The fluctuation and multiple sources of generated data combined with the dynamic utility-driven pricing model of cloud resource make it a very challenging problem.

The inter-dependency between multiple stages of distributed computation, such as the interplay between the Map phase and the Reduce phase of Map Reduce programs, further escalates the complexity of the data movement; resource provisioning and final reduce selection problems in geo-distributed datacenters. In this paper, we address the problem of efficient scheduling with the goal of high performance, high availability and cost minimization by balancing five types of cost between the two Map Reduce phases across multiple geo-distributed datacenters: bandwidth cost, storage cost, computing cost, migration cost, and latency cost. Contributions: The major contributions of this work are summarized as follows: • We propose a framework that can

systematically handle the issues of data movement, resource provisioning as well as reducer selection under the context of running Map Reduce across multiple datacenters, and VMs of different types and dynamic prices.

We design an efficient and distributed online algorithm-Mini BDP that is able to minimize the long-term time-averaged operation cost. We formally analyze the performance of Mini BDP in terms of cost optimality and worst case delay. We show that the algorithm approximates the optimal solution within provable bounds and guarantees that the data processing can be completed within pre-defined delays.

We conduct extensive experiments to evaluate the performance of our online algorithm with real world datasets. The experiments result demonstrate its effectiveness as well its superiority in terms of cost, system stability and decision-making time to existing representative approaches (e.g., the combinations of data allocation strategies (proximity-aware, load balance-aware) and the resource provisioning strategies(e.g., stable strategy, heuristic strategy).

II. Literature Survey

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

"Dynamic request redirection and resource provisioning for cloud-based video services under heterogeneous environment", Cloud computing provides a new opportunity for Video Service Providers (VSP) to running compute-intensive video applications in a cost effective manner. Under this paradigm, a VSP may rent virtual machines (VMs) from multiple geo-distributed datacenters that are close to video requestors to run their services. As user demands are difficult to predict and the prices of the VMs vary in different time and region, optimizing the number of VMs of each type rented from datacenters located in different regions in a given time frame becomes essential to achieve cost effectiveness for VSPs. Meanwhile, it is equally important to guarantee users' Quality of Experience (QoE) with rented VMs. In this paper, we give a systematic method called Dynamical Request Redirection and Resource Provisioning (DYRECEIVE) to address this problem. We formulate the problem as a stochastic optimization problem and design a Lyapunov optimization framework based online algorithm to solve it. Our method is able to minimize the long-term time average cost of renting cloud resources while maintaining the user QoE.

"i2mapreduce: Incremental mapreduce for mining evolving big data," As new data and updates are constantly arriving, the results of data mining applications become stale and obsolete over time. Incremental processing is a promising approach to refresh mining results. It utilizes previously saved states to avoid the expense of re-computation from scratch. In this paper, we propose i^2MapReduce, a novel incremental processing extension to Map Reduce. Compared with the state-of-the-art work on Incoop, i^2MapReduce (i) performs key-value pair level incremental processing rather than task level re-computation, (ii) supports not only one-step computation but also more sophisticated iterative computation, and (iii) incorporates a set of novel techniques to reduce I/O overhead for accessing preserved fine-grain computation states.

"Cross-cloud map reduce for big data," Map Reduce plays a critical role as a leading framework for big data analytics. In this paper, we consider a geo distributed cloud architecture that provides Map Reduce services based on the big data collected from end users all over the world. Existing work handles Map Reduce jobs by a traditional computation-centric approach that all input data distributed in multiple clouds are aggregated to a virtual cluster that resides in a single cloud. Its poor efficiency and high cost for big data support motivate us to propose a novel data-centric architecture with three key techniques, namely, cross-cloud virtual cluster, data-centric job placement, and network coding based traffic routing. Our design leads to an optimization framework with the objective of minimizing both computation and transmission cost for running a set of Map Reduce jobs in geo-distributed clouds.

"Asymptotic scheduling for many task computing in big data platforms," Paper deals with the problem of scheduling a set of jobs across a set of machines and specifically analyzes the behavior of the system at very high loads, which is specific to Big Data processing. We show that under certain conditions we can easily discover the best scheduling algorithm, prove its optimality and compute its asymptotic throughput. Present a simulation infrastructure designed especially for building/analyzing different types of scenarios. This allows extracting scheduling metrics for three different algorithms (the asymptotically optimal one, FCFS and a traditional GA-based algorithm) in order to compare their performance. Focus on the transition period from low incoming job rates load to the very high load and back. Interestingly, all three algorithms experience a poor performance over the transition periods. Since the Asymptotically Optimal algorithm makes the assumption of an infinite number of jobs it can be used after the transition, when the job buffers are saturated.

III. Problem Definition

Big data era with more data generated and collected in a geographically distributed manner in many areas such as finance, medicine, social web, astronomy etc. Square Kilometer Array an international project to build the world's largest telescope distributed over several countries, need to fusion the geographically dispersed data for scientific applications. However, due to the properties such as large-scale volume, high complexity and depressiveness of big data coupled with the scarcity of Wide area bandwidth (e.g., trans-oceanic link), it is inefficient and/or infeasible to process the data with centralized solutions.

IV. System Implementation

In this paper, the challenges (how to allocate data among a selection of geo-distributed datacenters to reduce the communication cost, how to determine the VM (Virtual Machine) provisioning strategy that offers high performance and low cost) are addressed by balancing bandwidth cost, storage cost, computing cost, migration cost, and latency cost, between the two MapReduce phases across datacenters.

We formulate this complex cost optimization problem for data movement, resource provisioning and reducer selection into a joint stochastic integer nonlinear optimization problem by minimizing the five cost factors simultaneously. The Lyapunov framework is

integrated into our study and an efficient online algorithm that is able to minimize the long-term time-averaged operation cost is further designed. Theoretical analysis shows that our online algorithm can provide a near optimum solution with a provable gap and can guarantee that the data processing can be completed within pre-defined bounded delays. Experiments on WorldCup98 web site trace validate the theoretical analysis results and demonstrate that our approach is close to the offline-optimum performance and superior to some representative approaches.

V. Online Control Algorithm Design

An outstanding feature of Lyapunov optimization is that it does not need future information about workload. By greedily minimizing the drift-plus-penalty at each time slot, it can solve the long-term optimization problem efficiently with a solution that can be proved to arbitrarily close to the optimum. Next, we first transform the problem P1 to an optimization problem of minimizing the Lyapunov drift-plus-penalty term and then design the corresponding online algorithm.

Fortunately, a careful investigation of the R.H.S (Right Hand Side) of inequality reveals that the optimization problem can be equivalently decoupled into three sub problems: 1) data allocation, 2) resource provisioning and (3) reducer selection. The details of solving the three subproblems are given as follows.

The three complex problems of data allocation, resource provisioning and reducer selection at time slot t have been solved independently and efficiently. The simple strategies facilitate the online deployment of the algorithm in the real-world systems. Employing the queue updating manner along time slots, we can design an online algorithm called *MiniBDP* for solving the problems in the long run.

VI. Procedure Definitions

Online Algorithm *MiniBDP*

Step1: *Resource provisioning*

The left part of R.H.S related to variable $m^k_d(t)$ (t) and $n^k_d(t)(t)$ can be considered as resource provisioning problem if we remove the constant term. The resource provisioning in each datacenter are independent, similar to data allocation, problem can be solved distributed within each datacenter.

Step2: Data Allocation

To minimize the R.H.S, and observing the relationship among variables, the part related to data allocation can be extracted from the R.H.S. Furthermore, since the data generated from each location are independent. The centralized minimization can be implemented independently and distributed.

Step2: Reducer Selection

Select the reducer to which aggregates the intermediate data from Map phase (i.e., $xd(t)$ is obtained). Update the queues $Md(t)$; $Yd(t)$;$Rd(t)$;$Zd(t)$ according to queue dynamic.

VII. Performance Analysis

In the experiments, two metrics cost and queue size are mainly considered. Cost measures the economic aspects of the system while queue size describes the stability of the system. To facilitate the comparison, Cost Ratio (*CR*), this measures the cost proportion of a single case among the total cost obtained by all cases.

Performance under Fixed Setting

In this section, we conducted a group of experiments under fixed parameters to evaluate the effectiveness of *MiniBDP*. Bellow presents the total cost of the system incurred over time slots.

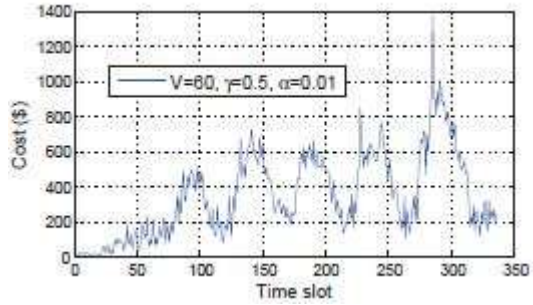

Figure: Cost incurred over time slots

MiniBDP is able to adaptively lease and adjust VMs resources to meet dynamic data processing demands even in a flash-crowded style without forecasting the future workload information. This is substantially different from those existing methods that need prediction phase in algorithm design. When we done with comparisons among different VM types, from which we find that the larger the VM capacity is, the more the number of VM with the corresponding type will be rented. This is probably because we design the pricing strategy with the principle that the more capacity of the VM is, the lower the unit price of the VM is. Thus, the algorithm prefers to rent the VM instance with large capacity (e.g.,c3.8xlarge) to process the data. The cost components processing cost, storage cost, bandwidth cost, latency cost, and migration cost are each time slot are also compared. This shows that processing cost occupies the major part of the total cost and the other types of cost are relatively low. This reveals that the algorithm is able to select the suitable datacenter for data processing while reducing the extra cost.

IX. Comparisons

In this section, we compare *MiniBDP* with other alternatives, each of which is the combination of a data allocation strategy, VM provisioning strategy and reducer selection strategy. For the data allocation strategies, three approaches are considered. 1) Proximity-aware Data Allocation (*PDA*), in which dynamically generated data are always allocated to the spatially nearest datacenter. It produces minimal latency and is suitable for the scenario that latency delay is prior to other factors. 2) Load-balancing Data Allocation (*LBDA*), in which the data are always allocated to the datacenter with the minimal workload. Obviously, this strategy is capable of keeping workload balanced among datacenters. 3) Minimal Price Data Allocation (*MPDA*), in which all the data are allocated to the datacenter with the lowest price at current time slot, so as to achieve the lowest cost.

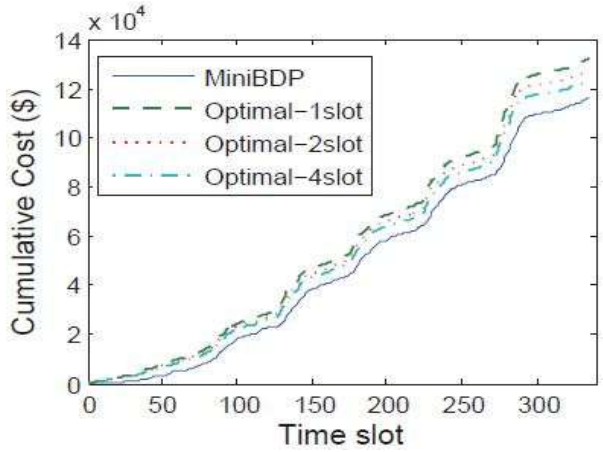

Figure: Cumulative cost comparison

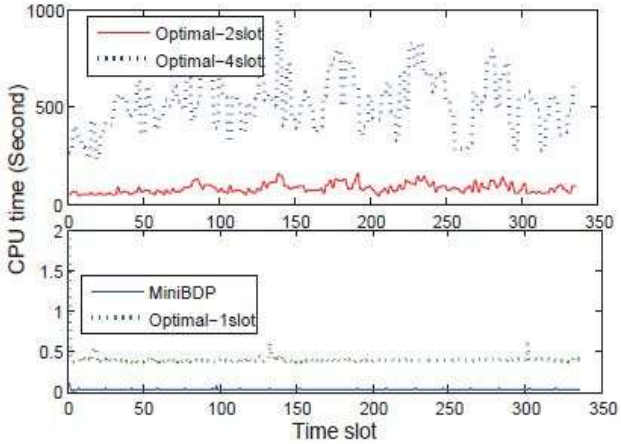

Figure: Solving time comparison

For the VM provisioning strategies, two typical strategies are considered. 1) Heuristic VM Provisioning (*HVP*), in which the VMs at current time slot are provisioned according to the workload at previous time slot. To better cope with the workload fluctuation, we add 50 percent VMs to those required at previous time slot to form the provisioned VMs. 2) Stable VM Provisioning (*SVP*), in which the VM count of each type in each datacenter is fixed a long time slots. For comparison, we set the average number of each VM type achieved by our algorithm as the stable resource provisioning strategy. In the whole sense, their total number of VMs of each type consumed are equal.

For the reducer selection strategies, we consider two approaches as follows. 1) Minimal Migration Cost Reducer Selection (*MCRS*), this takes the migration cost priority to select the reducer. 2) Load Balance Reducer Selection (*LBRS*), which selects the datacenter with the smallest workload of Reduce as the reducer.

X. Conclusion

Thus in this paper, a methodical framework for effective data movement, resource provisioning and reducer selection with the goal of cost minimization is developed. We balance five types of cost: bandwidth cost, storage cost, computing cost, migration cost, and latency cost, between the two MapReduce phases across datacenters. This complex cost optimization problem is formulated into a joint stochastic integer nonlinear optimization problem by minimizing the five cost factors simultaneously. By employing Lyapunov technique, we transform the original problem into three independent sub problems that can be solved by designing an efficient online algorithm MiniBDP to minimize the long-term time-average operation cost. The proposed approach is predicted to be with widespread application prospects in those globally-serving companies since analyzing the geographically dispersed datasets is an efficient way to support their marketing decision. As the subproblems in the algorithm MiniBDP are with analytical or efficient solutions that guarantee the algorithm running in an online manner, the proposed approach can be easily implemented in the real system to reduce the operation cost.

XI. References

[1] "Square kilometre array," http://www.skatelescope.org/.
[2] A. Vulimiri, C. Curino, B. Godfrey, T. Jungblut, J. Padhye, and G. Varghese, "Global analytics in the face of bandwidth and regulatory constraints," in Proceedings of the USENIX NSDI'15, 2015.
[3] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.
[4] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in Proceedings of the USENIX HotCloud'10, 2010.
[5] E. E. Schadt, M. D. Linderman, J. Sorenson, L. Lee, and G. P.Nolan, "Computational solutions to large-scale data management and analysis," Nature Reviews Genetics, vol. 11, no. 9, pp. 647–657, 2010.
[6] M. Cardosa, C. Wang, A. Nangia et al., "Exploring mapreduce efficiency with highly-distributed data," in Proceedings of the second international workshop on MapReduce and its applications, 2011.
[7] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. C. M. Lau, "Moving big data to the cloud: An online cost-minimizing approach," IEEE Journal on Selected Areas in Communications, vol. 31, pp. 2710–2721, 2013.
[8] W. Yang, X. Liu, L. Zhang, and L. T. Yang, "Big data real-time processing based on storm," in Proceedings of the IEEE TrustCom'13, 2013.
[9] Y. Zhang, S. Chen, Q. Wang, and G. Yu, "i2mapreduce: Incremental mapreduce for mining evolving big data," IEEE Transactions on Knowledge and Data Engineering, vol. 27, pp. 1906–1919, 2015.
[10] D. Lee, J. S. Kim, and S. Maeng, "Large-scale incremental processing with mapreduce," Future Generation Computer Systems, vol. 36, no. 7, pp. 66–79, 2014.
[11] B. Heintz, A. Chandra, R. K. Sitaraman, and J. Weissman, "End-toend optimization for geo-distributed mapreduce," IEEE Transactions on Cloud Computing, 2014.
[12] C. Jayalath, J. Stephen, and P. Eugster, "From the cloud to the atmosphere: Running mapreduce across data centers," IEEE Transactions on Computers, vol. 63, no. 1, pp. 74–87, 2014.
[13] P. Li, S. Guo, S. Yu, and W. Zhuang, "Cross-cloud mapreduce for big data," IEEE Transactions on Cloud Computing, 2015, dOI:10.1109/TCC.2015.2474385.
[14] A. Sfrent and F. Pop, "Asymptotic scheduling for many task computing in big data platforms," Information Sciences, vol. 319, pp. 71–91, 2015.
[15] L. Zhang, Z. Li, C. Wu, and M. Chen, "Online algorithms for uploading deferrable big data to the cloud," in Proceedings of the IEEE INFOCOM, 2014, pp. 2022–2030.
[16] Q. Zhang, L. Liu, A. Singhand et al., "Improving hadoop service provisioning in a geographically distributed cloud," in Proceedings of IEEE Cloud'14, 2014.

[17] A. Vulimiri, C. Curino, P. B. Godfrey, K. Karanasos, and G. Varghese, "Wanalytics: Analytics for a geo-distributed data-intensive world," in Proceedings of the CIDR'15, 2015.
[18] K. Kloudas, M. Mamede, N. Preguica, and R. Rodrigues, "Pixida: Optimizing data parallel jobs in wide-area data analytics," Proceedings of the VLDB Endowment, vol. 9, no. 2, pp. 72–83, 2015.
[19] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl, and I. Stoica, "Low latency geo-distributed data analytics," in Proceedings of the ACM SIGCOMM'15, 2015.
[20] "Facebook's prism project," http://www.wired.com/wiredenterprise/2012/ 08/facebook-prism/.
[21] J. C. Corbett, J. Dean, M. Epstein et al., "Spanner: Google's globallydistributed database," in Proceedings of the OSDI'12, 2012.
[22] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936–1948, 1992.
[23] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely, "Dynamic resource allocation and power management in virtualized data centers," in Proceedings of the IEEE NOMS, 2010, pp. 479–486.
[24] F. Liu, Z. Zhou, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in saas clouds," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 10, pp. 2648–2658, 2014.