Data Locality-Aware Query Evaluation for Big Data Analytics in Distributed Clouds

Qiufen Xia, Weifa Liang and Zichuan Xu
Research School of Computer Science
Australian National University, Canberra, ACT 0200, Australia
Email: qiufen.xia@anu.edu.au, wliang@cs.anu.edu.au, edward.xu@anu.edu.au

***Abstract*—With more and more enterprises and organizations outsourcing their IT services to distributed clouds for cost savings, historical and operational data generated by these services grows exponentially, which usually is stored in the data centers located at different geographic location in the distributed cloud. Such data referred to as big data now becomes an invaluable asset to many businesses or organizations, as it can be used to identify business advantages by helping them make their strategic decisions. Big data analytics thus is emerged as a main research topic in distributed cloud computing. The challenges associated with the query evaluation for big data analytics are that (i) its cloud resource demands are typically beyond the supplies by any single data center and expand to multiple data centers; and (ii) the source data of the query is located at different data centers. This creates heavy data traffic among the data centers in the distributed cloud, thereby resulting in high communication costs. A fundamental question for query evaluation of big data analytics thus is how to admit as many such queries as possible while keeping the accumulative communication cost minimized. In this paper, we investigate this question by formulating an online query evaluation problem for big data analytics in distributed clouds, with an objective to maximize the query acceptance ratio while minimizing the accumulative communication cost of query evaluation, for which we first propose a novel metric model to model different resource utilizations of data centers, by incorporating resource workloads and resource demands of each query. We then devise an efficient online algorithm. We finally conduct extensive experiments by simulations to evaluate the performance of the proposed algorithm. Experimental results demonstrate that the proposed algorithm is promising and outperforms other heuristics.**

I. Introduction

Cloud computing has emerged as the main computing paradigm in the 21st century [1], [10], by providing a plethora of cloud services including photo-uploading, online shopping, and IT service outsourcing. Distributed clouds [1], [13], [14], [23], [24], consisting of multiple data centers located at different geographical locations and interconnected by high-speed communication links, are the major cloud platforms of globally competitive services. With more and more enterprises and organizations now outsourcing their IT services to distributed clouds for cost-savings, security, reliability, performance, etc, distributed clouds can not only meet ever-growing user demands but also be capable to store the daunting volume of user historical and operational data into their data centers. These data then becomes the most invaluable asset to many businesses and organizations. Query evaluation on such big data is crucial in helping various businesses and organizations make critical business decision and improve their productivities in this competitive world.

Query evaluation for big data analytics in a distributed cloud typically requires lots of computing, storage and communication resources across multiple data centers, this incurs a great communication cost among data centers, by replicating the source data of the query from the data centers they are originally stored to the data centers where the query will be evaluated. To efficiently evaluate a query for big-data analytics, two issues must be addressed. One is how to identify a set of data centers that have sufficient computing and storage resources to meet the resource demands of the query. Another is how to minimize the communication cost of the query evaluation since the amounts of available bandwidth between different data centers varies over time and the communication cost is expensive due to large quantities of data transfers during the query evaluation [1], [11].

To motivate our study, we here use an example to illustrate the query processing for big data analytics in a distributed cloud (see Fig. 1), where there is such a query, assume that its source data is located at two data centers, v_1 and v_2, respectively. A naive evaluation plan for it is to replicate its source data from one data center to another one, e.g., from v_1 to v_2, or from v_2 to v_1, and then evaluate the query in v_2 or in v_1, as shown in Fig. 1(a). This however may incur a very expensive cost on network resource, especially when the volume of data transferred is in the scale of terabytes or even petabytes whereas the geographic locations of v_1 and v_2 are far-away from each other. Furthermore, this evaluation plan may not be feasible if neither v_1 nor v_2 has enough available computing and storage resources to meet the resource demands of the query evaluation. A smarter solution to the problem is to find an ideal data center v_3 with sufficient computing and storage resources that is not far away from both of them as depicted in Fig. 1(b). However, it is very likely that this ideal data center does not exist when all data centers are working at their high workloads at this moment. To respond to the query on time, sometimes multiple data centers must be employed so that their aggregate available resources can meet the resource demands of the query (e.g., v_3 and v_4 are employed). Thus, the available communication bandwidth between v_3 and v_4, v_1 and v_4, and v_2 and v_3 become crucial to meet the SLA requirement (the response time requirement) of the query, as

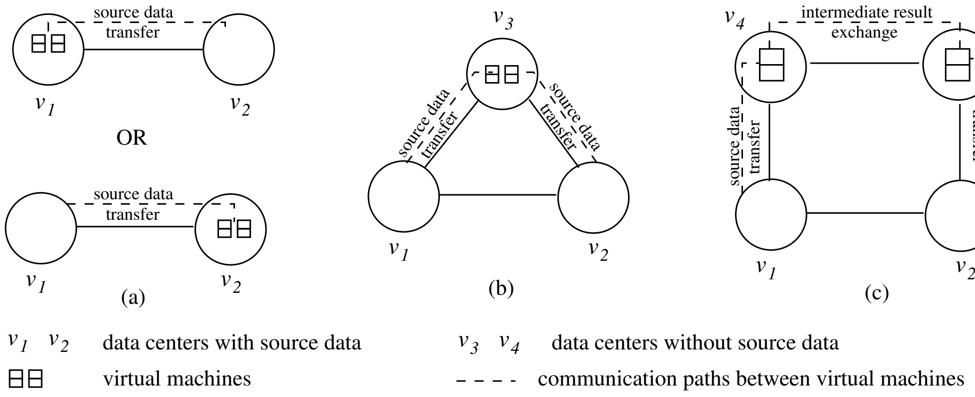


Fig. 1. A motivation example of query evaluation for big data analytics.

shown in Fig. 1(c). Motivated by this example, in this paper we deal with the online query evaluation problem for big data analytics in a distributed cloud. That is, for a given monitoring period, user queries for big data analytics arrive to the system one by one, the source data of each query is located at different data centers, the problem is to admit as many queries as possible (i.e., the query acceptance ratio) while keeping the accumulative communication cost of evaluating these admitted queries minimized.

Although extensive studies on query evaluation in cloud environments have been taken in the past several years [3], [7], [8], [13], [14], [15], [18], [20], [21], [25], most of them focused mainly on minimizing the computing cost [3], [14], [18], storage cost [18], or the response time in a single data center [15], little attention has been paid to the communication cost incurred when replicating source data and exchanging intermediate data among data centers, not to mention the impact of the source data locality on the cost of query evaluation. Although some of these studies [4], [6], [19], [21], [22], [25] considered the data locality issue, they focused only on a single data center, not multiple data centers located at different geographic locations. In contrast, in this paper we consider the query evaluation for big data analytics in a distributed cloud, one main character of this evaluation is the tight coupling between source data and computing resource demands of each query, since computing-intensive queries can be efficiently evaluated only when their source data are easily accessible and their computing resource demands are satisfied. Specifically, we deal with the online query evaluation problem in a distributed cloud by jointly considering source data localities, data center selections, source data replication and processing, and intermediate result exchanges, with an objective to maximize the query acceptance ratio while keeping the accumulative communication cost minimized. To the best our knowledge, very few studies considered online query evaluation for big data analytics, the work in this paper is the first one for the problem.

The main contributions of this paper are summarized in the following. We first propose a novel metric to model various resource utilizations in different data centers and the network resource of inter-data centers. We then devise an efficient online algorithm of query evaluation for big data analytics in distributed clouds. We finally conduct experiments by simulations to evaluate the performance of the proposed algorithm. Experimental results demonstrate that the proposed algorithm is very promising.

The reminder of this paper is organized as follows. Section II introduces the system model and the problem definition. The online evaluation algorithm for big data analytics is proposed in Section III. The performance evaluation of the proposed algorithm is given in Section IV, followed by introducing related work in Section V. The conclusion is given in Section VI.

II. Preliminaries

A. System model

We consider a distributed cloud $G = (V, E)$ consisting of a number of data centers located at different geographical locations and interconnected by high speed links, where V and E are the sets of data centers and links. Let v_i be a data center in V and $e_{i,j}$ a link in E between data centers v_i and v_j. Denote by $C(v_i)$ and $C(e_{i,j})$ the computing and bandwidth resource capacities of $v_i \in V$ and $e_{i,j} \in E$, respectively. Since the query evaluation for big data analytics usually is both compute and bandwidth intensive, the computing resource in data centers and the communication bandwidth resource on links between inter-data centers must meet the query resource demands. We here assume the computing resource demands of each query evaluation are given, represented by the number of VMs.

Assume that time is slotted into equal *time slots*, the resources in G are scheduled at each time slot. The amounts of available resources of $v_i \in V$ and $e_{i,j} \in E$ at different time slots may be significantly different, depending on the workload of that resource at that time slot. Denote by $A(v_i, t)$ the amount of the available computing resource in data center v_i and $B(e_{i,j}, t)$ the available communication bandwidth on link $e_{i,j}$ at time slot t. In this paper we focus only on the communication cost (bandwidth consumption) between data centers while ignoring the communication cost within each data center, as the former is treated as the bottleneck in query evaluation [5].

B. Query evaluation

Given a query Q for big data analytics with its source data located at multiple data centers in G, let $V_Q \subseteq V$ be the set of data centers at which the source data is located, and $S(v_i, Q)$ the size of the source data of Q at data center $v_i \in V_Q$. Assume that the number of VMs to evaluate Q is given in advance. In this paper, the evaluation of query Q consists of two stages: one is to identify a set V_P of data centers meeting the resource demands of Q; and another is to choose a subset $V_S \subseteq V_P$ of data centers such that the number of VMs contained by them is no less than the VM demands of Q. The *communication cost* of evaluating Q is the sum of the communication cost incurred by replicating its source data from the data centers in V_Q to the data centers in V_S and the communication cost between the data centers in V_S due to intermediate result exchanges.

To replicate the source data from a data center in V_Q to another data center in V_S or migrate intermediate results between two data centers in V_S, a routing path between the two data centers must be built. Let $p_{i,k}$ be a routing path in G between a pair of data centers v_i and v_k, the communication cost incurred by transferring source data $S(v_i, Q)$ from $v_i \in V_Q$ to $v_k \in V_S$ along path $p_{i,k}$ is $S(v_i, Q) \cdot c(p_{i,k})$, where $c(p_{i,k}) = \sum_{e \in p_{i,k}} c(e)$ is the cost of replicating a unit of data along $p_{i,k}$, and e is a link in $p_{i,k}$ [24], [25]. Notice that the choice of $p_{i,k}$ will be dealt with later. The intermediate results generated at each data center $v_k \in V_S$ may need to migrate to other data centers in V_S for the sake of the query evaluation. Let $I(v_k, Q)$ be the size of the intermediate result of Q in $v_k \in V_S$, the communication cost incurred is $(I(v_k, Q) + I(v_l, Q)) \cdot c(p_{k,l})$ by exchanging its intermediate result with the one in another data center $v_l \in V_S$ via a routing path $p_{k,l}$, where $c(p_{k,l}) = \sum_{e \in p_{k,l}} c(e)$ is the accumulative cost of replicating a unit of data via each edge $e \in p_{k,l}$. Denote by c_Q the communication cost of evaluating query Q, then,

$$c_Q = \sum_{v_i \in V_Q} \sum_{v_k \in V_S} S(v_i, Q) \cdot c(p_{i,k}) + \sum_{v_k, v_l \in V_S} (I(v_k, Q) + I(v_l, Q)) \cdot c(p_{k,l}), \tag{1}$$

where the first item in the right-hand side of Eq.(1) is the sum of communication costs between the data centers containing source data and the data centers processing query Q, while the second item is the communication cost among the data centers in V_S by exchanging intermediate results of Q.

For a given monitoring period of T that consists of T time slots. Let $\Delta Q(t)$ be the set of queries arrived between time slots $t-1$ and t, and $\Delta A(t)$ the set of admitted queries by the system at each time slot t, $1 \le t \le T$. Denote by $r(T)$ the *query acceptance ratio* for a period T, which is the ratio of the number of admitted queries to the number of arrived queries, i.e.,

$$r(T) = \frac{\sum_{t=1}^{T} |\Delta A(t)|}{\sum_{t=1}^{T} |\Delta Q(t)|}. \tag{2}$$

The *accumulative communication cost* of evaluating admitted queries for a period of T, $c(T)$, is thus

$$c(T) = \sum_{t=1}^{T} \sum_{Q \in \Delta A(t)} c_Q. \tag{3}$$

C. Problem definition

Given a distributed cloud $G = (V, E)$ for a monitoring period T, a sequence of queries for big data analytics arrive one by one without their arrival knowledge in future, assume that for each query Q in the sequence, the number of VMs required by it and its source data set V_Q $(\subseteq V)$ are given, the *source data locality-aware online query evaluation problem* in G for a monitoring period T is to deliver an query evaluation plan for each admitted query such that the query acceptance ratio $r(T)$ is maximized, while the accumulative cost $c(T)$ is minimized.

III. Query evaluation algorithm

In this section, we devise an efficient algorithm for the source data locality-aware online query evaluation problem for the monitoring period T. The algorithm proceeds in the beginning of each time slot t. For each arrival query $Q \in \Delta Q(t)$, the algorithm first checks whether the available VMs (computing resource) of data centers can meet its VM demands. If not, the query will be rejected; otherwise the query is processed. In the following, we deal with the evaluation of query Q in two stages (stage III-A and stage III-B).

A. Identification of a set V_P of potential data centers

To identify a set of data centers that meets the VM demands of query Q, a metric measuring the computing resource utilizations among data centers is needed. Such a metric should take into account not only the quantity of available computing resources but also their utilization ratios among data centers. Typically, the computing ability of a data center v_i decreases with the increase of its utilization ratio, the computing ability of a data center v_i thus is modelled as the *data center metric*, denoted by $\Phi(v_i, t)$ at time slot t, then

$$\Phi(v_i, t) = A(v_i, t) \cdot a^{\frac{A(v_i, t)}{C(v_i)}}, \tag{4}$$

where a is a constant with $a > 1$ that represents in which degree the resource utilization is, $A(v_i, t)$ is the amount of available computing resource of v_i. The ratio $\frac{A(v_i,t)}{C(v_i)}$ models the computing resource utilization of v_i. A higher $\Phi(v_i, t)$ means that v_i has more available computing resource and a lower resource utilization, and has a higher probability to be a potential processing data center for query Q. Similarly, the *link metric* $\Psi(e_{i,j}, t)$ of a link $e_{i,j}$ between two data centers v_i and v_j at time slot t is defined by

$$\Psi(e_{i,j}, t) = B(e_{i,j}, t) \cdot b^{\frac{B(e_{i,j}, t)}{C(e_{i,j})}}, \tag{5}$$

where $b > 1$ is a similar constant, $B(e_{i,j}, t)$ is the available bandwidth resource of link $e_{i,j}$ at time slot t. The arguments of ratio $\frac{B(e_{i,j},t)}{C(e_{i,j})}$ and $\Psi(e_{i,j}, t)$ are similar as we did for $\frac{A(v_i,t)}{C(v_i)}$ and $\Phi(v_i, t)$.

The allocated VMs for evaluating query Q require communications with each other in order to exchange their intermediate results. This can be implemented through building multiple routing paths between the data centers accommodating the VMs. To find a cheaper routing path p, the 'length' of path p is defined as the sum of lengths of links in it. Let $d(e,t)$ be the length of link e in p at time slot t, if $\Psi(e,t) > 0$, then $d(e,t) = \frac{1}{\Psi(e,t)}$; $d(e,t) = \infty$, otherwise. This implies that the shorter the length of a link, the more available the bandwidth on it.

Having defined the metrics of resources in data centers and on links, we now identify a set V_P of potential data centers to evaluate query Q. We first identify the 'center' of the set V_P by assigning each data center $v_i \in V$ *a rank* $NR(v_i,t)$ that is the product of data center metric $\Phi(v_i,t)$ of v_i and the accumulative metric of links incident to v_i, i.e.,

$$NR(v_i,t) = \Phi(v_i,t) \cdot \sum_{e_{i,j} \in L(v_i)} \Psi(e_{i,j},t), \tag{6}$$

where $L(v_i)$ is the set of links incident to v_i in G. The rationale behind Eq. (6) is the more available computing resources a data center v_i has, the more available accumulative bandwidth of links incident to it, and the higher rank the data center v_i will have. A data center with the highest rank will be selected as the 'center' of the set of data centers V_P, denoted by v_c.

If the available computing resource $\Phi(v_c,t)$ of v_c cannot meet the resource demands of query Q, the next data center will be chosen and added to V_P greedily. Specifically, each data center $v_i \in (V - V_P)$ is assigned a rank by the product of the inverse of $\Phi(v_i,t)$ and the accumulative shortest length from v_i to all the selected data centers $v_j \in V_P$, i.e.,

$$\frac{1}{\Phi(v_i,t)} \cdot \sum_{v_j \in V_P} \sum_{e_{i,j} \in p_{i,j}} d(e_{i,j},t), \tag{7}$$

where $p_{i,j}$ is the one with the minimum accumulative length of links, i.e., the path between $v_i \in V - V_P$ and each selected data center $v_j \in V_P$ is the shortest one. The data center with the smallest rank is chosen and added to V_P. This procedure continues until the accumulative computing resource of all chosen data centers in V_P meet the demanded number of VMs of query Q.

B. Selecting a subset V_S of V_P

Recall that the source data of query Q in data center $v_i \in V_Q$ will be replicated to a data center for the query evaluation, i.e., we assume that this data cannot be split and replicated to multiple data centers. Such an assumption is purely for the sake of simplicity of discussion, which can be easily extended to the multiple data center case. The second stage of the evaluation algorithm is to identify a subset V_S of V_P to minimize the communication cost between the data centers of source data and the data centers performing the query evaluation. To this end, we reduce this stage into an unsplitable minimum cost multi-commodity flow problem in an auxiliary directed flow graph $G' = (V', E')$ whose construction is as follows.

A *virtual sink node* t_0 and all data center nodes in G are added to G', i.e., $V' = V \cup \{t_0\}$. There is a directed link from each node $v_j \in V_P$ to the virtual sink node t_0. The capacity of edge e_{v_j,t_0} is the volume of source data that v_j can process at time slot t, and its cost is set to zero. For each pair of data centers, there are two directed edges between them in G' if there is an edge between them in G. The capacity of each edge in $E' \setminus \{\langle v_j, t_0 \rangle | v_j \in V_P, V_P \subseteq V\}$ is set to the total volume of source data of query Q, and the cost of each such edge is set to the communication cost by replicating a unit of source data along it.

The source data in each data center $v_i \in V_Q$ is treated as *a commodity* with demand $S(v_i,Q)$ at a source node in G', which will be routed to the destination node t_0 through a potential data center $v_j \in V_P$. To find a feasible solution in G', we first find a shortest routing path for each commodity in terms of its link costs. We then route the commodity to a data center that has the minimum ratio of the cost of its minimum-cost path to its source data size. This procedure continues until all commodities are routed successfully. The detailed algorithm is described in **Algorithm 1**.

Theorem 1: Given a distributed cloud $G = (V,E)$ for a given monitoring period of T time slots, assume that queries for big data analytics arrive one by one without future arrival knowledge. There is an online algorithm for the source data locality-aware online query evaluation problem, which takes $\sum_{t=1}^{T} O(|\Delta Q(t)| \cdot (|V|^3 \log |V| + \cdot |V|^2 \cdot |E|))$ time, where a set $\Delta Q(t)$ of queries arrived at each time slot t, $1 \le t \le T$.

Proof: Following **Algorithm** 1, consider each time slot t with $1 \le t \le T$, if a given query $Q \in \Delta Q(t)$ is admitted, then it takes $O(|V|^2)$ time to identify the set of data centers V_P with enough computing resources to meet the VM demands of Q in stage one. While in stage two, the dominant time spent on routing a commodity is to select a commodity $S(v_i,Q)$ with the minimum ratio of $\min_{v_i \in V} \frac{S(v_i,Q)}{\sum_{e \in p_{i,t_0}} c(e)}$, where p_{i,t_0} is the shortest path between v_i and t_0. This takes $O\big(|V|^2 \log |V| + |V| \cdot |E|\big)$ time to find all shortest paths in G for the $|V|$ commodities. The total amount of time for routing all commodities takes $O(|V| \cdot (|V|^2 \log |V| + |V| \cdot |E|)) = O\big(|V|^3 \log |V| + |V|^2 \cdot |E|\big)$, and there are $|\Delta Q(t)|$ queries, **Algorithm** 1 takes $O\big(|\Delta Q(t)| \cdot (|V|^3 \log |V| + \cdot |V|^2 \cdot |E|)\big)$ time per time slot. The total amount of time of the online algorithm for the monitoring period T thus is $\sum_{t=1}^{T} O\big(|\Delta Q(t)| \cdot (|V|^3 \log |V| + \cdot |V|^2 \cdot |E|)\big)$. ■

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm and investigate the impact of different parameters on the algorithm performance.

A. Simulation environment

We consider a distributed cloud $G(V,E)$ consisting of 20 data centers, there is an edge in E between a pair of nodes with a probability of 0.2 generated by the GT-ITM tool [12]. The computing capacity of each data center and the bandwidth capacity of each link are randomly drawn from value intervals

Algorithm 1: Algorithm for the locality-aware dynamic query evaluation problem

```
Input: The distributed cloud graph $G = (V, E)$, a set $T$ of queries
       $\Delta Q(t)$, the set of VMs to evaluate each query $Q \in \Delta Q(t)$ at
       time slot $t$.
Output: The query acceptance ratio and the accumulative
        communication cost of evaluating admitted queries.
1  for $t \leftarrow 1, 2, ..., T$ do
2      $Q \in \Delta Q(t)$ is the query being evaluated;
3      if the available computing resources can not meet the resource
       demands of $Q$ then
4          $Q$ is rejected;
5      end
6      else
           // Stage III-A
7          $V_p \leftarrow \emptyset$. Calculate rank $NR(v_i, t)$ for each data center
           $v_i \in V$ according to Eq. (6), and select the data center with
           the maximum rank, i.e., $v_c$. $V_P \leftarrow V_P \cup \{v_c\}$;
8          while data centers in $V_P$ do not have enough computing
           resource for the VMs needed by Q, and $V \setminus V_p \neq \emptyset$ do
9              Find $v_i \in V \setminus V_p$, with minimum value of Eq. (7),
               $V_p \leftarrow V_P \cup \{v_i\}$;
10         end
           // Stage III-B
11         Create an auxiliary graph $G' = (V', E')$, $E' \leftarrow E$, and
           $V' \leftarrow V \cup t_0$, where $t_0$ represents a virtual destination for all
           commodities of $Q$, i.e., $S(v_i, Q), \forall v_i \in V_Q$;
12         for each potential data center $v_j \in V_P$ do
13             Add an edge $e_{j,t_0}$ from $v_j$ to $t_0$. The cost of $e_{j,t_0}$ is 0,
               and its capacity is $A(v_j, t)/R_m \cdot S_m$;
14         end
15         for each edge $e$ in $E'$ do
16             The capacity of $e$ is set to the total volume source data
               of query $Q$;
17             The cost of $e$, $c(e)$, is set to the communication cost for
               replicating one unit of source data of $Q$;
18         end
19         for each commodity $S(v_i, Q)$ do
20             Find a path $p_{i,t_0}$ from $v_i$ to $t_0$ with the minimum
               accumulated cost of all edges along the path;
21             Calculate the ratio $\sum_{e \in p_{i,t_0}} c(e)/S(v_i, Q)$;
22         end
23         Route the commodity with the minimum ratio
           $\sum_{e \in p_{i,t_0}} c(e)/S(v_i, Q)$, delete this commodity, and update
           capacities and costs of edges in $G'$;
24         Repeat steps 19 to 23 until no commodities can be routed;
25         if there are still commodities unrouted but $V \setminus V_p = \emptyset$ then
26             $Q$ is rejected;
27         end
28         else
29             Add the data center in $V \setminus V_p$ with minimum value of
               Eq. (7) into $V_P$ and repeat procedures from step 12 to
               step 27 until all commodities are routed;
30             Update available computing resources of data centers and
               bandwidth resources of links in $G$;
31         end
32     end
33     $\Delta Q(t) \leftarrow \Delta Q(t) \setminus \{Q\}$;
34 end
35 Return query acceptance ratio and the accumulative communication
   cost of evaluating admitted queries.
```

$[1,000,\ 3,000]$ units (GHz), and $[100,\ 1,000]$ units (Mbps), respectively [2], [9]. The total volume of source data of each query is in the range of [128, 512] GB, which are randomly distributed between 1 and 4 data centers. Each virtual machine has the computing capacity of 2.5 GHz and can process 256 MB data chunk [17], according to the settings of Amazon EC2 Instances [1]. Parameters a and b are set as 2^4 and 2^6 in default settings. We assume that the monitoring period T is 800 time slots with each time slot lasting 5 minutes. We further assume that the number of query issued within each time slot is ranged between 5 and 45, and each query evaluation takes between 1 and 10 time slots. According to the on-demand pricing of Amazon CloudFront, users pay only for the contents that are delivered to them through the network without minimum commitments or up-front fees, and the fee charged for transmitting 1 GB data is in the range of [\$0.08, \$0.12]. Unless otherwise specified, we will adopt these default settings. Each value in the figures is the mean of the results by applying the mentioned algorithm 15 times.

To evaluate the proposed algorithm, two heuristics are used as evaluation baselines. One is to choose a data center with the maximum number of available VMs and then replicate as much source data of the query as possible to the data center. If the data center cannot meet the query resource demands, it then picks the next data center with the second largest number of available VMs. This procedure continues until the VM demands of the query are met. Another is to select a data center randomly and places as much source data of the query as possible to the data center. If the available number of VMs in the chosen data center is not enough to process the query, it then chooses the next one randomly. This procedure continues until the VM demands of the query are met. For simplicity, we refer to the proposed algorithm and the two baselines algorithms as `DL-Alg`, `Greedy-Alg`, and `Random-Alg`, respectively.

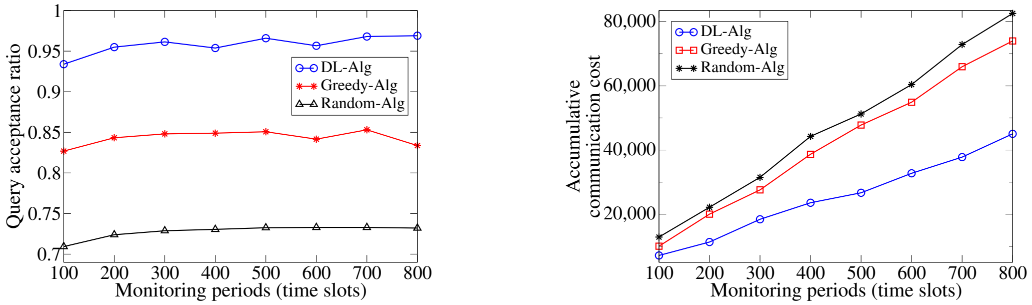
B. Algorithm performance evaluation

We first evaluate the performance of the proposed algorithm in terms of the query acceptance ratio and accumulative communication cost. Fig. 2(a) plots the curves of query acceptance ratios by three mentioned algorithms `DL-Alg`, `Greedy-Alg` and `Random-Alg`, from which it can be seen that the query acceptance ratio by algorithm `DL-Alg` is much higher than that of algorithm `Random-Alg`, which is about 11% and 22% higher than those of algorithms `Greedy-Alg` and `Random-Alg` respectively, while Fig. 2 (b) shows the accumulative communication cost of these three algorithms. Clearly, the accumulative communication costs of algorithms `Greedy-Alg` and `Random-Alg` is worsen than that by algorithm `DL-Alg`, which are around twice that by algorithm `DL-Alg`.

C. Impact of parameters on algorithm performance

We then study the impact of the number of data centers n on the query acceptance ratio and accumulative communication cost by varying the number from 10 to 40.

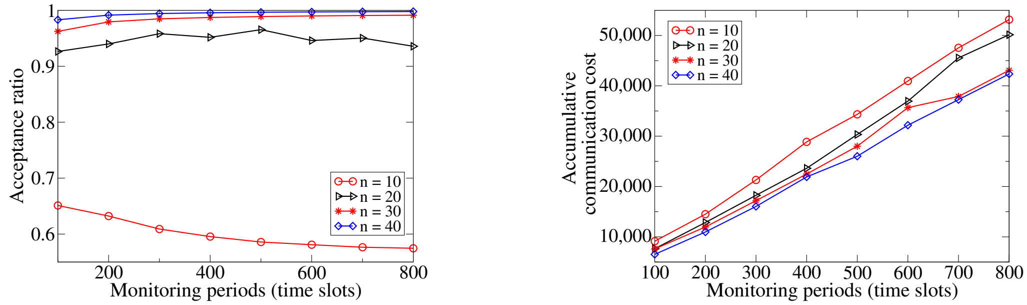
Fig. 3(a) plots the query acceptance ratio curves of algorithm `DL-Alg`, from which it can be seen that the query acceptance ratio first grows with the increase of n, and then keeps stable after $n = 30$. For example, the acceptance ratio grows by 18% when the value of n increases from 10 to 20. The reason is that with the increase of the number of data centers, more and more queries are admitted by the system as more computing resources now are available, and the query



(a) The query acceptance ratio of algorithms `DL-Alg`, `Greedy-Alg` and `Random-Alg`

(b) The accumulative communication cost of algorithms `DL-Alg`, `Greedy-Alg` and `Random-Alg`

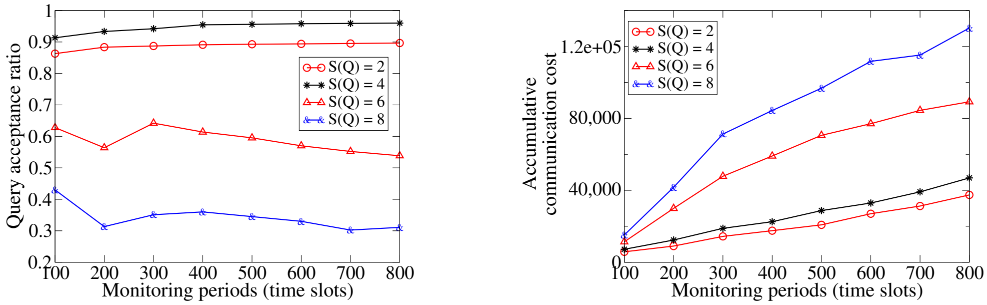
Fig. 2. Performance evaluation of different algorithms for various monitoring periods.



(a) The impact on the query acceptance ratio

(b) The impact on the accumulative communication cost

Fig. 3. Impacts of the number of data centers on the performance of algorithm `DL-Alg` over various monitoring periods.



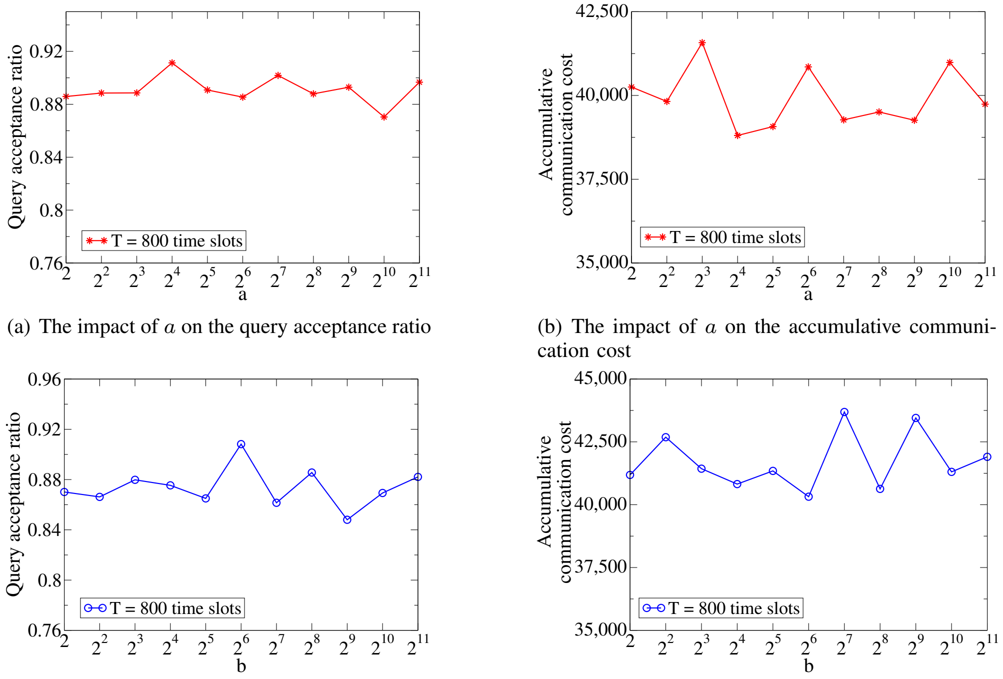
(a) The impact on acceptance ratio

(b) The impact on accumulative communication cost

Fig. 4. The impact of source data location numbers on the performance of algorithm `DL-Alg` over different monitoring periods.

acceptance ratio approaches 100% when $n = 30$ due to the abundant computing resource available. Fig. 3(b) depicts the accumulative communication cost curves of algorithm `DL-Alg`, which decreases, with the increase of the number of data centers n.

We thirdly evaluate the impact of the maximum number of data sources of each query on the query acceptance ratio and accumulative communication cost by varying it from 2 to 8. Fig. 4 is the result chart, where $S(Q)$ is employed to represent the maximum number of source data locations of query Q. Fig. 4(a) plots the query acceptance ratio curve by algorithm `DL-Alg`, from which it can be seen that the query acceptance ratio grows first and then decreases with the increase of $S(Q)$. Specifically, the query acceptance ratio increases gradually from 88% to 94% when $S(Q) = 2$ and $S(Q) = 4$, and then decreases to 55% and 30% when $S(Q) = 6$ and $S(Q) = 8$. The rationale behind is that, fewer data sources of a query means that it has larger volume of source data at each of source data centers, if its total volume of source data is given. This may also increase the probability of query rejection due to lack of computing resources. However, if the number of data sources $|S(Q)|$ is quite high (i.e., the source data of the query are distributed more data centers) then this implies that more frequent source data replication and intermediate results exchange are needed. Such queries also tend to be rejected by the system as limited bandwidth resources imposed between data centers. Fig. 4(b) plots the accumulative communication cost curve of algorithm `DL-Alg`. It can be seen that the accumulative communication cost increases with the growth of $|S(Q)|$. For example, when $|S(Q)|$ is 8, the accumulative



(a) The impact of a on the query acceptance ratio

(b) The impact of a on the accumulative communication cost

(c) The impact of b on the query acceptance ratio

(d) The impact of b on the accumulative communication cost

Fig. 5. The impact of a and b on the query acceptance ratio and the accumulative communication cost of algorithm `DL-Alg` under $T = 800$ time slots.

communication cost is 3.1 times, 2.8 times, and 1.5 times of that when $|S(Q)|$ are 2, 4, and 6. The reason is that more data sources may lead to more frequent source data replication and intermediate results exchange, thereby consuming more communication bandwidth among data centers.

We finally investigate the impact of the parameters a and b on the query acceptance ratio and accumulative communication cost by varying them from 2^1 to 2^{11} when $T = 800$. From Fig. 5(a), it can be seen that the query acceptance ratio reaches the peak when $a = 2^4$ and then decreases when $a > 2^4$. The rationale behind is that when $a < 2^4$, the data center metric of each data center v_i, $\Phi(v_i, t)$, is not large enough to impact its ranking in the selection of processing data centers, which results in that the algorithm may choose some data centers with less available computing resources, thereby increasing the rejection probability of queries with large volume of source data. On the other hand, when $a > 2^4$, the algorithm may select data centers whose incident links have less available bandwidth resources, since the data center metric dominants the ranking of data centers, queries with high communication demands then is tended to be rejected. Similar behavior patterns can be found in Fig. 5(b), omitted. Fig. 5(c) demonstrates that the acceptance ratio reaches its peak when $b = 2^6$ and then decreases when $b > 2^6$. The rationale is that when $b < 2^6$, the link metric ($\Psi(e_{i,j}, t)$) of the incident links of a data center is too low to dominate the ranking of the data center. This may lead to some data centers with insufficient network resources on their incident links are selected for the query, thereby increasing the rejection probability of queries with more communication demands. In contrast. when $b > 2^6$, the opposite may happen, the ranking of a data center will be dominated by its incident link metric $\Psi(e_{i,j}, t)$, and the selected data centers may reject some queries as they may not have enough available computing resources. Similarly, it can be seen from Fig. 5(d) the accumulative communication cost reaches its minimum when $b = 2^6$. Thus, b is set at 2^6 in the default setting.

V. Related Work

Several studies have investigated query evaluation in clouds in recent years [18], [15], [3], [14], [8], [7], [20], [13], [25]. For example, Mian *et al.* [18] examined the query evaluation problem of provisioning resources in a public cloud by selecting a configuration for the query such that the sum of computing and storage costs of the configuration is minimal, assuming that all data accessed by the query is local data. Kllapi *et al.* [15] provided a distributed query processing platform, Optique, to reduce the query response time. However none of them considered the communication cost of query evaluation, which in fact cannot be ignored due to the migration of massive data and the limited bandwidth among servers in a data center. Bruno *et al.* [3] proposed an optimization framework for continuous queries in cloud-scale systems. Particularly, they continuously monitored the query execution, collected runtime statistics and adopted different execution plans during the query evaluation. If a new plan is better than the current one, they will adopt the new plan with minimal costs. A similar problem was studied in [14], the only difference [14] lies in a small sample of data drawn for query execution to estimate the cost of the query evaluation plans. However, providing an accurate execution plan is time-consuming due to the massive data analysis in cloud-scale data centers, and suboptimal plans can be disastrous with large datasets. Liu *et al.* [16] proposed three information retrieval for ranked query schemes to reduce the query overhead incurred on the cloud. They assumed that users can choose the query rankings to determine the percentage of matched results to return. To this end, a mask matrix is used to filter out a certain percentage of matched results, i.e., their primary motivation is providing the users a scheme to restrict the number of results. Unfortunately, the correct filtering and the returned amounts of results are difficult to decide.

There are other studies that focused mainly on minimizing the computing cost [18], [3], [14], [22], the storage cost [18], and/or the query response time [15]. Little attention has ever been paid to the communication cost due to source data and immediate results migrations in a distributed cloud. Also, source data localities is an important issue which impacts the cost of query evaluation. Although several recent papers [19], [21], [22] considered data locality when dealing with query evaluation, they focused only on one single location (data center). For example, Tung *et al.* [21] investigated the query evaluation on databases, each of which is represented by a rooted, edge-labeled directed graph, i.e., a distributed graph. They assumed that all related data are sent to one data center for processing, whereas the cloud resources in this data center may not meet the resource demands of the query. Different from these existing studies, in this paper we consider the query evaluation for big data analytics in a distributed cloud consisting of multiple data centers. The main feature of this study is how to tightly couple between the source data and the computing resource demands of each query, since the source data of each query usually are located in multiple data centers, and the query can be efficiently evaluated only when its source data is easily accessible and its computing resource demands can be met. We term this problem as the online query evaluation problem by jointly considering source data localities, query evaluation data center selections, source data replication, and intermediate evaluation results exchanges between processing data centers, with an objective to maximize the query acceptance ratio while keeping the accumulative communication cost minimized. To the best our knowledge, very few studies on query evaluation for big data analytics in distributed clouds have ever considered.

VI. Conclusion

In this paper, we considered an online query evaluation problem for big data analytics in distributed clouds with an objective to maximize the query acceptance ratio while keeping the accumulative communication cost minimized. We first proposed a novel metric model to model various cloud resource utilizations of different data centers by incorporating different resource workloads among the data centers and the resource demands of each query. We then devised an efficient online algorithm for the problem. We finally conducted extensive experiments by simulation to evaluate the performance of the proposed algorithm. Experimental results demonstrate that the proposed algorithm is promising and outperforms two mentioned heuristics.

References

[1] Amazon EC2. `http://aws.amazon.com/ec2/instance-types/`.

[2] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron. Towards predictable datacenter networks. *Proc.of SIGCOMM*, ACM, 2011.

[3] N. Bruno, S. Jain, and J. Zhou. Continuous cloud-scale query optimization and processing. *J. Proceedings of the VLDB Endowment*, Vol.6, pp.961-972, 2013.

[4] X. Bu, J. Rao, and C. Xu. Interference and locality-aware task scheduling for MapReduce applications in virtual clusters. *Proc. of HPDC*, ACM, 2013.

[5] Y. Chen, S. Jain, V. K. Adhikari, Z. Zhang, and K. Xu. A first look at inter-data center traffic characteristics via Yahoo! datasets. *Proc. of INFOCOM*, IEEE, 2011.

[6] M. Cardosa, C. Wang, A. Nangia, A. Chandra, and J. Weissman. Exploring MapReduce efficiency with highly-distributed data. *Proc. of MapReduce*, ACM, 2011.

[7] D. J. Abadi. Data management in the cloud: limitation and opportunities. *J. IEEE Data Eng. Bull.*, Vol.32, pp.3-12, 2009.

[8] W. Fan, X. Wang, and Y. Wu. Performance guarantees for distributed reachability queries. *Proc. of the VLDB Endowment*, ACM, 2012.

[9] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang. SecondNet: a data center network virtualization architecture with bandwidth guarantees. *Proc. ACM CONEXT*, 2010.

[10] Google Cloud Platform. `https://cloud.google.com/`

[11] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *J. SIGCOMM Computer Communication Review*, Vol.39, pp.68-73, ACM, 2009.

[12] `http://www.cc.gatech.edu/projects/gtitm/`.

[13] L. Gu, D. Zeng, P. Li, and S. Guo. Cost minimization for big data processing in geo-distributed data centers. *IEEE Trans. on Emerging Topics in Computing*, preprint.

[14] K. Karanasos, A. Balmin, M. Kutsch, F. Ozcan, V. Ercegovac, C. Xia, and J. Jackson. Dynamically optimizing queries over large scale data platforms. *Proc. of SIGMOD*, ACM, 2014.

[15] H. Kllapi, D. Bilidas, I. Horrocks, Y. Ioannidis, E. Jim nez-Ruiz, E. Kharlamov, M. Koubarakis, and D. Zheleznyakov. Distributed query processing on the cloud: the Optique point of view (short paper). *OWL Experiences and Directions Workshop*, 2013.

[16] Q. Liu, C. C. Tan, J. Wu, and G. Wang. Towards differential query services in cost-efficient clouds. *Trans. on Parallel and Distributed Systems*, Vol.25, pp.1648-1658, IEEE, 2014.

[17] Understanding MapReduce input VS. file chunk sizes. `https://www.mapr.com/developercentral/code/understanding-mapreduce-input-vs-file-chunk-sizes#.VDcolKVlnng`.

[18] R. Mian, P. Martin, and J. L. Vazquez-Poletti. Provisioning data analytic workloads in a cloud. *J. Future Generation Computer Systems*, Vol.29, pp.1452-1458, Elsevier, 2013.

[19] B. Palanisamy, A. Singh, L. Liu and B. Jain. Purlieus: Locality-aware resource allocation for MapReduce in a cloud. *Proc. of SC*, ACM, 2011.

[20] S. Sakr, A. Liu, D.M. Batista, and M. Alomari. A survey of large scale data management approaches in cloud environments. *J. Communications Surveys and Tutorials*, Vol.13, pp.311-336, IEEE, 2011.

[21] L. Tung, Q. Nguyen-Van, and Z. Hu. Efficient query evaluation on distributed graphs with Hadoop environment. *Proc. of SoICT*, ACM, 2013.

[22] S. Wu, F. Li, S. Mehrotra, and B. C. Ooi. Query optimization for massively parallel data processing. *Proc. of SOCC*, ACM, 2011.

[23] Z. Xu and W. Liang. Minimizing the operational cost of data centers via geographical electricity price diversity. *Proc. of Cloud*, IEEE, 2013.

[24] L. Zhang, Z. Li, C. Wu, and M. Chen. Online algorithms for uploading deferrable big data to the cloud. *Proc. of INFOCOM*, IEEE, 2014.

[25] 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. *J. on Selected Areas in Communications*, Vol.31, pp.2710–2721, IEEE, 2013.