Trusted Support Service and OTP method for Moving Big Data to Cloud

[1]Muhsina.B.S,[2]PreethikaRajashekhar,[3]Supriya.V,[4]ThusleenaKabeer,[5]Navin K.S
LBS Institute of Technology for Women, Thiruvananthapuram, Kerala, India muhsinabs@gmail.com,
[1]preethz2408@gmail.com, [2]suppuvl@gmail.com, [3]thas235@gmail.com, [4]navinsivendran@gmail.com

Abstract: **Cloud computing is a mechanism in which company hires the data centre and hosts the confidential information for computing the various services in low cost infrastructure, on demand and dynamic provisioning. An important open issue here is to efficiently move the data, from different geographical locations over time, into a cloud for effective processing. The de facto approach of hard drive shipping is not flexible or secure. This work studies timely, cost-minimizing upload of massive, dynamically-generated, geo-dispersed data into the cloud, for processing using a Map Reduce-like framework. But it outstands with security concerns. Security requirements for cloud computing environment should have trusted computing platform. The system proposes cloud computing system is combined with Trusted Support Service (TSS). In this design, better user authentication can be obtained by using One Time Password (OTP) which will sent to user via email and SMS. The system also provides the Trusted Support Service (TSS) by using Java/C editor. Paper also demonstrates the performance comparison of this system. For cost-minimizing data migration problem, we propose two online algorithms: an online lazy migration (OLM) algorithm and a randomized fixed horizon control (RFHC) algorithm , for optimizing at any given time the choice of the data centre for data aggregation and processing, as well as the routes for transmitting data there. Here we are demonstrating this concept through a banking application**

Keywords: **Cloud Computing, Big Data, and Online algorithms**

I. INTRODUCTION

Cloud computing is a powerful technology to perform massive-scale and complex computing. It eliminates the need to maintain expensive computing hardware, dedicated space, and software. The elastic and on- demand nature of resource provisioning makes a cloudplatform attractive for the execution of various applications, especially computation-intensive ones [1], [2], [3]. More and more data-intensive Internet applications, *e.g.*, Facebook, Twitter, and big data analytics applications, such as the Human Genome Project [4], are relying on the clouds for processing andanalyzing their petabyte-scale datasets, with a computing framework such as MapReduce andHadoop.Massive growth in the scale of data or big data generated through cloud computing has been observed. Addressing big data is a challenging and time- demanding task that requires a large computational infrastructure to ensure successful data processing and analysis

While most efforts have been devoted to designing bettercomputing models for big data analytics, an important issuehas largely been left out in this respect. The current practice to movethe massive amounts of data into a cloud, is to copy the data into largehard drives for physically transportation to the data center or even to move entire machines. Such physicaltransportation incurs undesirable delay and possible servicedowntime, while outputs of the data analysis are often neededto be presented to users in the most timely fashion . It is alsoless secure, given that the hard drives are prone to infectionof malicious programs and damages from road accidents. Asafer and more flexible data migration strategy is in need, tominimize any potential service downtime.

The challenge escalates when we target at dynamically and continuously produced data from different geographical locations. Withdynamic data, an efficient *online algorithm* is desired, fortimely guiding the transfer of data into the cloud over time.For geo- dispersed data sets, we can select the best datacenter to aggregate all data onto, for processing with aMapReduce-like framework, which is efficient to process datawithin *one* data center but not across data centers, due to theenormous overhead of inter-data center data moving in thestage of shuffle and reduce Two efficient online algorithms are proposed to practicallyguide data migration over time: an *online lazy migration(OLM)* algorithm and a *randomized fixed horizon control(RFHC)* algorithm. Theoretical analyses show that the OLMalgorithm achieves a worst-case competitive ratio of 2.55,without the need of any future information and regardlessof the system scale, under the typical settings in real-worldscenarios. The RFHC algorithm achieves a competitive ratio of $1 + (1\ l+1)(\ \kappa\ \lambda)$that approaches 1 as the lookahead window l grows. Here κ and λ are system dependent parameters ofsimilar magnitudes.

For better user authentication we propose OTP in our system. This authentication system will ensure authorized access toinformation building trust in cloud users. The proposedsystem application will be Integrated DevelopmentEnvironment (IDE) for Java/C source code compilation,debugging and execution facilities. Only registered users can use this cloud service.To ensure information security, system will implement emailOTP authentication technique. For every login, an OTP willbe sent to registered email id/mobile no. The cloud user has tofetch the OTP to email account/mobile and then only he canaccess cloud services like Java/C source compilation,debugging and execution.
In the rest of the paper, we discuss related work in Sec. II, describe the system and problem models in Sec. III, and present the online solutions in Sec. IV,Sec V describes the security methods that can be provided and Sec. 6 concludes the paper.

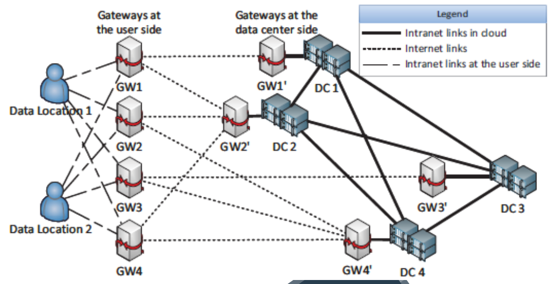
II. RELATED WORK

A series of recent work studies application migration to the cloud. Hajjat et al. [5] develop an optimization model for migrating enterprise IT applications onto a hybrid cloud. Cheng et al. [6] and Wu et al. [7] advocate deploying social media applications into clouds, for leveraging the rich resources and pay-as you-go pricing. These projectsfocus mainly on workflow migration and application performance optimization, by carefully deciding the modules to be moved to the cloud and the data caching/replication strategies in the cloud. The very first question of how to move large volumes of applicationdata into the cloud is not explored. Few existing work discussed such transfer of big data to the cloud. Cho et al. [8] design Pandora, a cost-aware planning system for data transfer to the cloud provider, via both the Internet and courier services.. Different from our study, they focus on static scenarios with a fixed amount of bulk data to transfer, rather than dynamically generated data; in addition, a single cloud site is considered, while our study pays attention to multiple data centers.

III. THE DATA MIGRATION PROBLEM

A. System Model

We can consider a cloud consisting of K geo-distributed data centers in a set of regions.A cloud user continuously produces large volumes of data at a set *D* of multiple geographic locations. The user connects to the data centers from different data generation locations via virtual private networks (VPNs), with *G*



VPN gateways at the user side and K VPN gateways each collocated with a data center.

B.*Cost-minimizing Data Migration: Problem Formulation*

Assume the system executes in a time-slotted fashion with slot length τ. Fd(t) bytes of data are produced at location d in slot t, for upload to the cloud. ldg is the latency between data location d ∈D and user side gateway g ∈G , pgi is the delay along VPN link (g,i), and ηik is the latency between data centersi and k. These delays, which can be obtained by a simple command such as ping, are dictated by the respective geographic distances. A cloud user needs to decide(i) via which VPN connections to upload its data to the cloud, and (ii) to which data center to aggregate data, for processing by a MapReduce-like framework, such that the monetary charges incurred, as well as the latency for the data to reach the aggregation point, are jointly minimized. The total cost C to be minimized has four components: bandwidth cost, aggregate storage and computing cost, migration cost, and routing cost.

Decisionvariables. Two types of optimization variables are formulated:

(1) Data routing variable xd,g,i,k(t),∀d ∈D ,∀g ∈G ,∀i∈ K, ∀k ∈K, denotes the portion of data Fd(t) produced at location d in t, to be uploaded through VPN connection (g,i) and then migrated to data center k for processing.

xd,g,i,k(t) > 0 indicates that the data routing path d → g → i → k is employed, and xd,g,i,k =0 otherwise.
Let $\vec{x}$ =(xd,g,i,k(t))∀d∈D,∀g∈G,∀i∈K,∀k∈K, the set of feasible data routing variables are:
X= { ($\vec{x}$(t)) | g∈G,i∈K,k∈KΣxd,g,i,k(t)=1and xd,g,i,k∈ [0,1], ∀d ∈D ,∀g ∈G ,∀i∈K ,∀k ∈K} (1)
HereΣg∈G,i∈K,k∈Kxd,g,i,k(t)=1ensures that all data produced from location d are uploaded into the cloud in t.
(2) Binary variable yk(t), ∀k ∈K, indicates whether data center k is target of data aggregation in time slot t (yk(t)= 1) or not (yk(t)=0). Following the practical requirement of the current MapReduce framework, we require that at any given

time, exactly one data center is chosen. Let $y^{\rightarrow}(t)$ = (yk(t))∀k∈K, the set of possible data aggregationvariablesare:

Y={ (y⃗(t)) |Σk∈Kyk(t)=1andyk(t) ∈{ 0,1},∀k∈K}(2)

Costs. The costs incurred in time slot t, for uploading the data into the cloud and for processing the data at the selected data center, include the following components.

(1)*The overall bandwidth cost* for uploading data via the VPN connections,where d∈D,k∈KΣFd(t)xd,g,i,k(t) is the amount uploaded via (g,i), and fgi is the charge for uploading one byte of data via(g,i), derived from bandwidth prices set by the cloud provider:

CBW(x⃗(t))≜Σg∈G,i∈K (fgiΣ d∈D,k∈KFd(t)xd,g,i,k(t)). (3)

(2)*Storage and computation costs* are important factors to consider in choosing the data aggregation point.The data produced in t and also from the past which is in the form of raw data or intermediate processing results may involve in processing and analysing.Without loss of generality, let the amount of current and history data to process in t be F(t)=Σ*tv*=1(αvΣd∈DFd(v)), whereΣd∈DFd(v) is the total amount of data produced in time slot v from different data generation locations, and weight αv∈[0,1] is smaller for older times v and αt =1for the current time t. Specific applications determines the value of αv and it can be obtained through statistical data . Assume all the other historical data, except those inF(t), are removedfrom the data centers where they were processed, since all needed information has been stored in the retained data. Let Ψk(F(t)) be a non- decreasing cost function for storage and computation in data centerk in t. The aggregate storage and computing cost incurred in the cloud in t is:

$C_{DC}(y^{\rightarrow}(t)) \triangleq \Sigma_{k\in K} y_k(t)\ \Psi_k(F(t))$. (4)

The best data center for data aggregation can differ in t than in t−1, due to temporal and spatial variations in data generation.Historicaldata neededfor processing togetherwith new data in t, at the amount of Σ*t*−1*v*=1(αvΣd∈DFd(v)), should be moved from the earlier data center to the current, and a *migration cost* is incurred. Let φik(z) be the non-decreasing migration cost to move z bytes of data from data centeri to date center k. The migration cost between t−1 and t is:

$C_{tMG}(y^{\rightarrow}(t),y^{\rightarrow}(t-1)) \triangleq \Sigma_{i\in K}\Sigma_{k\in K}([y_i(t-1)-y_i(t)]^+$

$[y_k(t)-y_k(t-1)]^+\varphi_{ik}(\Sigma t-1v=1\alpha_v\Sigma_{d\in D}F_d(v)))$ (5)

(4) The latency incurred for data upload is an important performance measure, to be minimized in the data routing and aggregation process. Targeting both monetary cost minimization and delay minimization, a routing cost for delays along the selected routing paths is formulated, and combine it with other costs to be the optimization objective. The overall routing cost in the system in t is:

CRT(x(t))≜d,g,i,k Lxd,g,i,k(t)Fd(t)(ldg + pgi + ηik). (6)

where xd,g,i,k(t)Fd(t)(ldg + pgi + ηik) is the product of data volume and delay along the routing path d → g → i → k. The weighted formula suggests that transferring a large volume of data via a high latency path causes high cost. L is the routing cost weight converting xd,g,i,k(t)Fd(t)(ldg+pgi+ηik) into a monetary cost, reflecting how latency-sensitive the user is. A cloud user specifies L as a constant a priori. Latency ldg +pgi +ηik is fixed in each slot but can change over time. In summary, the overall cost incurred in t in the system is:

C(x⃗(t),y⃗(t))=CBW(x⃗(t))+CDC(y⃗(t)) + CtMG (y⃗(t),y⃗(t-1)) + CRT(x⃗(t)). (7)

IV. TWO ONLINE ALGORITHMS

A. The Online Lazy Migration (OLM) Algorithm

We divide the overall costC(x⃗(t),y⃗(t)) incurred in t into two parts: (i) migration cost CtMG(y⃗(t), y⃗(t-1)) defined earlier, related to decisions in t−1; (ii) non- migration cost that relies only on current information at t:

Ct−MG(x⃗(t),y⃗(t))=CBW(x⃗(t))+CDC(y⃗(t)) + CRT(x⃗(t)). (8)

We design a lazy migration algorithm (Alg. 1), whose basic idea is to postpone data center switching even if the one-shot optimum indicates so, until the cumulative non-migration cost (in Ct−**MG**(x⃗(t),y⃗(t))) has significantlyexceeded the potential data migration cost.

Algorithm 1: The Online Lazy Migration (OLM) Algorithm

```
1: t =1 ;
2: t̂=1; //Time slot when the last change of aggregation data center happens
3: Compute data routing decision x⃗(1) and aggregation decision y⃗(1) by minimizing C(x⃗(1),y⃗(1))
4: Compute CMG 1(y⃗(1),y⃗(0)) and C−MG 1 (x⃗(1),y⃗(1));
5: while t ≤ T do
6:ifC−MG t̂(y⃗(t̂),y⃗(t̂−1))≤ 1β2ΣCv −MG(x⃗(v),y⃗(v))t−1v=t̂
then
7: Derive x⃗(t) and y⃗(t) by minimizing C−MG t (x⃗(t),y⃗(t)) and constraint CMG t (y⃗(t),y⃗(t-1)) ≤β1C−MG t (x⃗(t),y⃗(t));
8: if y⃗(t) ≠y⃗(t−1) then
9: Usethenew aggregation datacenter indicatedby y⃗(t);
10: t̂= t;
11: if t̂<tthen //not to use new aggregation data center 12: y⃗(t)=y⃗(t−1), compute data routing decision x⃗(t) by solving (10) if not derived;
13: t = t +1
```

At the beginning (t=1), we solve the one-shot optimization and upload data via the derived optimal routes $\vec{x}(1)$ to the optimal aggregation data center indicted by $\vec{y}(1)$. Let $t̂$ be the time of the data center switch. In each following time slot t, we compute the overall non-migration cost in [$t̂$,t − 1],Σt−1v=$t̂$Cv−**MG**($\vec{x}$(v),$\vec{y}$(v))). The algorithm checkswhether this cost is at least β2 times the migration cost CMG $t̂$($\vec{y}$($t̂$), $\vec{y}$($t̂$−1)). If so, it solves the one-shot optimizationto derive $\vec{x}$(t) and $\vec{y}$(t) withoutconsidering the migrationcost, i.e., by minimizing C−MG t($\vec{x}$(t),$\vec{y}$(t)) and an additional constraint, that the potential migration cost,C−MG t ($\vec{y}$(t),$\vec{y}$(t-1)), is no larger than β1 times the nonmigration cost C−MG t ($\vec{x}$(t),$\vec{y}$(t)) at time t (to make sure that the migration cost is not too excessive). If a change of migration data center is indicated ($\vec{y}$(t) = $\vec{y}$(t − 1)), the algorithm accepts the new aggregation decision, and migrates data accordingly.In all other cases, the aggregationdata center remainsunchangedfromt−1,whileoptimaldataroutingpaths are computed given this aggregation decision, for upload of new data generated in t.

Alg.1 avoids aggressive switches of the aggregation data center, to prevent moving a large amount of data back and forth too often. Excessive "laziness" is also avoided. Parameters β2 > 0 and β1 > 0 control the "laziness" and "aggressiveness" of the algorithm: a large β2 prolongs the inter-switch interval, while a large β1 invites more frequent switches.

B. The Randomized Fixed Horizon Control (RFHC) Algorithm

In practical applications, near-term future data generationpatterns can often be estimated from history.We assume that the information in the lookaheadwindow can be predicted precisely without error.We divide time into equal-size frames of l + 1 time slotseach (l ≥ 0). In the first time slot t of each frame, assumeinformation on data generation for the next l time slots, $i.e.$,Fd(t), Fd(t + 1), ..., Fd(t + l), ∀d ∈ D, are known. We solvethe following cost minimization over time frame [t, t + l],given the data aggregation decision of $\vec{y}$(t − 1), to derivedata routing decisions $\vec{x}$(v) and aggregation decisions $\vec{y}$(v),∀v = t, . . . , t + l, using Alg. 1

minimizeΣC($\vec{x}$(v),$\vec{y}$(v))t+lv=t. (9)

We design a Randomized Fixed Horizon Control (RFHC)algorithm (Alg. 2). At the beginning, the algorithm uniformlyrandomly chooses p ∈ [1, l + 1] as the start of the firsttime frame of l + 1 slots, $i.e.$, it randomly picks one specificalgorithm FHC(p)from the l+1 FHC algorithms: at t = 1, it solves (16) to decide the optimal data routing and aggregationstrategies in the period of t = 1 to p − 1 (p ≠1); then at$t$ = p, p + l + 1, p + 2(l + 1), . . ., it solves (16) for optimalstrategies in the following l + 1 time slots, respectively.

Algorithm 2: The Randomized Fixed Horizon Control (RFHC) Algorithm

```
1: $\vec{y}(0) = 0$;
2: $p = rand(1, l + 1)$; //A random integer within [1,l+1]
3: if $p \neq 1$ then
4: Derive $\vec{x}(1) \cdot\cdot\cdot \vec{x}(p - 1)$ and $\vec{y}(1) \ldots \vec{y}(p - 1)$ by solving(16) over the time window $[1, p - 1]$;

5: $t = p$;
6: while $t \leq T$ do
7: if $(t - p) \bmod (l + 1) = 0$ then
8: Derive $\vec{x}(t), \cdot\cdot\cdot, \vec{x}(t + l)$ and $\vec{y}(t), \cdot, \vec{y}(t + l)$ bysolving (16) over the time frame $[t, t + l]$;
9: $t = t + 1$;
```

An adversary, with no information on p, finds it hard to contrivespecific inputs to degrade the performance of RFHC.

V. SECURITY

In our proposed system, security can be ensured by two methods.

(A). One Time Password (OTP)

Security requirements for cloud computingenvironment should have trusted computing platform.Our proposed system will demonstrate One TimePassword through email to ensure information security. Thesystem will generate this OTP every time the user is trying tologin. The One time password will be mailed /sent SMS tothem for every login.

1.Implementing trusted computing platform

Dynamic password is generated either in time based or event base mechanisms. One-time password authentication system (OTP) provides authentication for system access (login).

Algorithm 3: OTP generation

```
Input: User's detailed information including Gmail account& mobile number
Output: One time password
1. Generate random values between 0-123(excluding 0,123)
2. Check the values in range of 47-57(0-9), 65-91(A to Z),97-122(a to z)
3. If the number is in the range then convert the number in to its ASCII value Else convert the number to its nearest number then convert the number in to its ASCII value
4. Append the generated characters in the password String(OTP)
```

(B).Privacy Preservation

Privacy preservation is an important issue in the release of data for mining. We focus on a study on the k-anonymity property[10]. The k-anonymity model assumes a *quasi-identifier*, which is a set of attributes that may serve as an identifier in the data set. It is assumed that the dataset is a table and that each tuple corresponds to an individual. Let Q be the quasi-identifier. An equivalence class of a table with respect to Q is a collection of all tuples in the table containing identical values for Q. The size of an equivalence class indicates the strength of identification protection of individuals in the equivalent class. If the number of tuples in an equivalence class is greater, it will be more difficult to re-identify individual. A data set D is k-anonymous with respect to Q if the size of every equivalence class with respect to Q is k or more. As a result, it is less likely that any tuple in the released table can be linked to an individual and thus personal privacy is preserved.

We propose a simple and effective model to protect both identifications and sensitive associations in a disclosed data set. The model extends the k-anonymity model to the (α,k)anonymity model to limit the confidence of the implications from the quasi-identifier to a sensitive value (attribute) to within α in order to protect the sensitive information from being inferred by strong implications.

DEFINITION: ((α,k)-ANONYMIZATION). A view of a table is said to be an (α,k)-anonymization of the table if the view modifies the table such that the view satisfies both k-anonymity and α-deassociation properties with respect to the quasi-identifier.

The approaches mentioned in this section solve the privacy issues while moving big data to cloud. (α,k)anonymity model protects sensitive attribute when the attribute contains many values and no single value dominates the attribute. At the same time user authentication is provided by the OTP method.

CONCLUSION

This paper designs efficient algorithms for timely, cost minimizing migration of geo-dispersed big data to thecloud, for processing using a MapReduce-like framework. Two online algorithms are designed to practically migrate data in an online fashion. Data security and privacy is one of the biggest challenges in cloud computing .Cloud data must be protected not only against external attackers but also corrupt insiders .Our proposed system uses OTP and Privacy Preservation approach which aims to make cloud data self intelligent.

REFERENCES

[1]. “Moving Big Data to The Cloud: An Online Cost-Minimizing Approach” Linquan Zhang, Chuan Wu, Zongpeng Li, Chuanxiong Guo, Minghua Chen, and Francis C.M. Lau

[2]. M. Armbrust, A. Fox, R. Grifth, A. D. Joseph, R. Katz, A. Konwinski,G. Lee, D. P. A. Rabkin, I. Stoica, and M. Zaharia, “Above the Clouds:A Berkeley View of Cloud Computing,” EECS, University of California,Berkeley, Tech. Rep., 2009.

[3]. S. Pandey, L. Wu, S. Guru, and R. Buyya, “A Particle Swarm Optimization(PSO)-based Heuristic for Scheduling Workflow Applicationsin Cloud Computing Environment,” in *Proc. IEEE AINA*, 2010

[4].*Human Genome Project*, http://www.ornl.gov/hgmis/home.shtml.

[5]. M. Hajjat, X. Sun, Y. E. Sung, D. Maltz, and S. Rao, “CloudwardBound: Planning for Beneficial Migration of Enterprise Applications tothe Cloud,” in *Proc. ACM SIGCOMM*, August 2010.

[6] X. Cheng and J. Liu, “Load-Balanced Migration of Social Media toContent Clouds,” in *Proc. ACM NOSSDAV*, June 2011.

[7] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. Lau, “Scaling Social MediaApplications into Geo-Distributed Clouds,” in *Proc. IEEE INFOCOM*, Mar. 2012.

[8] B. Cho and I. Gupta, “New Algorithms for Planning Bulk Transfer viaInternet and Shipping Networks,” in *Proc. IEEE ICDCS*, 2010.

[9]. Trusted Platform for support services in Cloud Computing environment SnehaKolhe, SudhirDhage*2

[10].Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu and Ke Wang,”(α,k)-Anonymity:An enhanced k-anonymity model for privacy preserving data publishing,”