DATA PLACEMENT AND CACHE MANAGEMENT SCHEME FOR CLOUD DATA SERVICES

V. Manimaran\textsuperscript{1}, R. Ramesh Kumar\textsuperscript{2}
\textsuperscript{1}M.E. (CSE),
\textsuperscript{2}Assistant Professor,
Department of Computer Science and Engineering
Gnanamani College of Technology
manimarankv@gmail.com

ABSTRACT--Data sharing is managed by the cloud service providers. Data availability and time constrained delivery are managed by the data caching schemes. Latency minimization and network traffic reduction factors are used to improve the quality of cloud services. Cost analysis is performed to manage the transmission and cache process. Data staging and caching operations are handled with cost factors. Data staging process handles the process of data placement under cloud sites. Data caching is performed to improve the data transmission process. Single copy and multiple copy models are used to manage the transmission and caching cost. Upper bound is assigned to control cost under multiple copy based model. Number of service requests and the number of distinct data items are used in the upper bound estimation process. Homogeneous and heterogeneous cost models are used in the data staging process. Optimal data staging algorithm and greedy algorithm are used in the system. The heterogeneous cost model is enhanced with single and multiple copy models. Item based cost model is improved to support bulk data transfer based cost assignment process. Dynamic caching cost optimization is provided in the system. Time dependant cache management cost estimation is improved in the system.

INTRODUCTION

The data staging or caching algorithms in different context generally focus on the first issue to judiciously predict or select the data items and/or the vantage staging sites to optimize various performance metrics. Some of them take advantages of the properties of the network graph while others are based on the historical data trace or the modeled access distribution. However, few of them considered the second issue in the algorithms to efficiently migrate, replicate or cache the shared data items. This is partially because most existing staging or caching algorithms in general target toward a class of services such as web caching, memory paging and so on where the access time sequence to a predicted data subset is usually irrelevant to the applications, which is different to our research context where the
data service optimization for time-variant requests is the major concern.

**Overview**

Suppose there are $k$ distinct shared data items initially stored at one node, say $p_1$ and later migrated, replicated and cached in a fully connected network with $m$ nodes ($p_1, p_2, \ldots, p_m$) to serve a sequence of requests $\sigma \triangleq \sigma_1\sigma_2\ldots\sigma_n$ in which each $\sigma_i \triangleq (t_i, p_i, R_i), 1 \leq i \leq n$ is specified by the predicted access pattern and represents a request made for a data subset $R_i$ by node $p_i$ at time $t_i$. Therefore, have the complete knowledge of all such information as an input to our algorithms. Further, it assumed that there exists only one request per stage. However, for multiple-request case it is briefly discuss it as a supplemental result presented in Appendix A.

To satisfy a request for a particular data item, it is defined that the following primitive operations to perform on the cached data items, which may involve caching and transmission costs:

1) **retention**: cache the data item at a node $p_u$ from time $t_u$ to $t_v$ by paying $(t_v - t_u)S_u$, $S_u$ is the rate of caching cost at node $p_u$, $1 \leq u \leq m$.

2) **migration**: move the data item from a node $p_u$ to a node $p_v$ at a cost equal to the distance $C_{uv}$.

3) **replication**: copy the item to the request node $p_v$ from a node $p_u$ at a cost of $C_{uv}$.

4) **excursion**: satisfy the request at a node $p_v$ by using the copy at a node $p_u$ without migration at a cost of $E_{uv}$.

5) **creation/deletion**: create/delete the selected copies at some nodes without incurring any cost.

Usually replication cost and excursion cost are not identical. However, in Cloud systems such as GAE Cloud Computing, Amazon Web Services and Windows Azure Platform, the resources, especially the storage, are always charged based on the amount and duration that clients use, not including their initialization and finalization.

Each new request can trigger one or more operations, leading to the creation and deletion of the data items at arbitrary nodes. This procedure can be visually described by a space-time diagram in which the horizontal axis represents the time stages while the vertical axis represents the nodes, the requests are shown as circles, also referred as request points hereafter. In contrast, the points other than the request points at the same time stage are called dummy points. But for a dummy point, if there is a copy cached at the corresponding node, the dummy point is also named...
excursion point for that copy. Since there is at most one request at each time stage, It can always use the time-stage index along the x-axis of the space-time diagram to specify the corresponding request if no confusion is incurred.

A typical example of the space-time diagram is shown in Fig. 1 where the transmission of the shared data from one node to another at a stage is represented by a vertical directed arrow connecting two points on a stage between the respective nodes and the caching of the shared data at a node from one stage to another is represented by hasking the horizontal time line between corresponding stages at that node. The goal of the proposed algorithms is to produce a data staging scheme by combining these primitive operations to satisfy the request sequence in an efficient way while minimizing the total cost. The total cost for the staging scheme is

\[ 2S_1 + 4S_2 + S_5 + C_{12} + 2C_{25} + C_{24}. \]

Due to the discussed merits of the monetary cost, it is deliberately ignore modeling some properties and features of the network platform such as network traffic, bandwidth capacity, link latency and CPU power. Rather, it is assume that the network links are perfect so that the transmission delay of data items could be neglected in our model. On the first sight this zero-delay assumption is too strong to be practical in reality. However, it is achievable after a simple transformation without lose of accuracy.

In the homogeneous cost model, the transmission cost between any pair of nodes are identical, denoted as \( C \) while the caching cost at all sites are also identical, denoted as \( S \). Our interest is to satisfy the entire request sequence of length \( n \), each being made for at most \( k \) distinct data items. Suppose \( d^{lv}_j \) is the cost for transferring item \( j \) from point \( u \) to point \( v \), which in our context is formally defined as:

\[
d^{lv}_j = \begin{cases} (t_v - t_u)S + \mu_jC & \text{if } t_v \geq t_u \\ +\infty & \text{otherwise} \end{cases}
\]

if it is simply represent point \( u \) and \( v \) as \((p_u, t_u)\) and \((p_v, t_v)\) respectively. Here \( \mu_j = 1 \) if \( p_u \neq p_v \), otherwise \( \mu_j = 0 \).

For this optimization problem, it allow single and multiple copies for each item. One characteristic that distinguishes our problem from others, especially the well-known (off-line) \textit{k-server problem}, is that the copies can be created and destroyed automatically to minimize the total cost. In addition, it is also motivated to consider some practical constraints on the maximum numbers of transmissions and copies in a
scheduling interval. The maximum numbers can be given either on per-item basis or on all-item basis. Under a heterogeneous cost model where each node $p_u$ has its own caching cost rate $S_u$, $1 \leq u \leq m$, and the transmission cost rate $C_{uv}$ between $p_u$ and $p_v$, $1 \leq u, v \leq m$ are not always identical, our problem has a similar formulation. The only difference is that the definition of $d_{uv}^j$ is the shortest path between $p_u$ and $p_v$ in the space-time diagram.

The data staging algorithm is periodically scheduled to update the access patterns whereby to achieve a new data staging and caching scheme based on the actual requests in a previous time interval. The scheduling interval could be a day, a week or other time unit.

**Objectives**

In this system, it does not intend to improve the first phase of the algorithm for a high-quality access pattern. Instead, with a such kind of information presumably known in advance, our goal is to efficiently migrate, replicate or cache the multiple requested data items defined by the access pattern in a fully connected network with or without resource constraints so that the sequence of service demands are satisfied with minimum data transmission and caching costs.

Achieving this goal will have two immediate advantages, which are critical to the success of applying such techniques to the maximization of data availability. First, as the cost in the data staging and caching accounts for a part of the total cost, minimizing it can accordingly minimize the total cost. Second, with the cost of the second phase decreasing, the first phase of the algorithm could have more opportunities to adjust the frequency of running the prediction algorithms to improve the prediction accuracy so that the overall cost could be reduced.

**SYSTEM STUDY**

**Existing System**

In this system, data staging problem by leveraging the dynamic programming (DP) techniques to optimally migrate, replicate and cache the shared data items in Cloud systems with or without resource constraints in an efficient way while minimizing the monetary cost for transmitting and caching the data items. Monetary cost is our primary interest as on one hand, monetary cost is a very flexible concept to reflect the qualities of various network features such as network bandwidth, link latency and storage utilization, and on the other hand, the
provision of the resources in Cloud systems are usually based on pay-as-you-go fashion, and thus effective use of the platforms within budget constraint is always the user’s concern.

Due to the optimality, our solutions are unique and advantageous over other methods to provide the Cloud-based services with the flexibility that they can not only decide the duration of each data item to be cached at some vantage sites but also make a trade-off between transmission cost and caching cost to meet the constraints imposed by the underlying IaaS/SPs, information item owners or CSPs’ budget. As cloud computing is gaining its prominence, these benefits are more important than ever before to the success of the traditional network-based services migrating to Clouds such as the distributed collaborative document editing process and multimedia personalized services where a document or video clip may be requested by users in a sequence of pre-defined time instants.

In general, data staging algorithm in our applications is not only needed to figure out the user’s access patterns but also required to migrate, replicate or cache the involved data items across the network at particular time instants to optimize some performance metric for future accesses. It is explicitly distinguish the two functional phases of the data staging algorithm with a burning focus on the second one. The first phase takes as input a variety of kinds of data such as a network graph, a historical data trace or other modeled access distributions. The output of this phase is an inferred access pattern which is taken as an input to the second phase. The output of the second phase is the total cost to achieve the data staging which in our particular case is the total monetary cost, a major concern in Cloud computing. Distinguishing between these two phases is reasonable as the functionality of each phase is different, so is the measure metric. Additionally, another apparent advantage is that the prediction/selection element of the algorithm is separated from the migration/replication element. Consequently, the two elements could be designed and implemented independently and their combinations could be compose-able to achieve the best performance.

The cost model adopted in our research could be heterogeneous or homogeneous in the senses that whether or not the transmission costs are identical and caching costs at all sites are also identical. As our algorithms are mainly designed for CSPs who usually demand the infrastructure
services from an IaaS provider, in this system it is particularly interested in the situation when the homogeneous cost model is employed due to two reasons: First, the rented infrastructure by CSPs for a particular service is always organized as a subset of homogeneous resources to entail the hosted applications to meet its SLA (Service Level Agreement) targets. This in general results in homogeneous computation and communication in the clouds.

- **Homogeneous Computation:** IaaS providers commonly provide users with a set of different virtual machine types, each of which has different resource capacities in terms of CPU capacity, RAM size, disk I/O bandwidth and the like. The performance of different types of virtual machines is obviously heterogeneous. However, the performance of multiple virtual machines of the same type which usually host a particular service is practically similar.

- **Homogeneous Communication:** the current topologies of datacenter networks are in general structured as either two- or three-level trees of switches or routers with a low-bandwidth edge tier at the leaves and high-bandwidth fat-tree at the top of the tree, roughly leading to homogeneous communication in nature.

Second, as both the transmission cost rate and caching cost rate are determined by the IaaS providers, it is unlikely for them to offer a heterogeneous cost model as it could pose difficulties in public clouds since at present only a few of IaaS providers are willing to expose some low-level information about the containers and sub-networks to their users.

Given multiple copies allowed for each data item, the situation when some restrictions are imposed on the volume of communications and the number of copies during a scheduling interval. These restrictions are rational in practice as they would be required either by some SLA, technical requirements or by some legal or economic reasons.

For example, the communication constraint is practical as in some current datacenters, to deliver full bandwidth between arbitrary nodes, multi-path routing protocols such as ECMP are commonly employed. Thus, limiting the number of transmissions in serving all the requests during a time interval can effectively prevent the communication fabric from being over-subscribed. On the other hand, some copyright laws require that a Cloud-
based service may be only allowed by the owner of a data item to make a specified number of copies in each defined time period for a fixed. As such a copy constraint needs to be defined for such a data item, and the Cloud-based service has to consider this constraint when staging the data item in the network with a minimum cost.

Overall, our network and cost models together with the resource constraints are reasonable and representative to conduct this research to reflect the reality. However, for completeness, the situation when the cost model is heterogeneous. To help understanding, the frequently used symbols in a notation table for quick reference in the following descriptions.

Under the homogeneous cost model, our results include two parts. First, when there are no resource constraints, DP-based optimal algorithm in polynomial time to stage and cache $k$ distinct items to satisfy a pre-determined sequence of requests, each item having single or multiple copies to minimize the cost. Based on this algorithm, it is also shown that the cost of a single copy algorithm is within a factor of $1 + \frac{C}{S}$ from that of the multi-copy algorithm. This result indicates that a single copy of each data item can efficiently serve all the user requests to it provided that the ratio of transmission cost rate (C) and caching cost (S) is low. Moreover, when $S > C$, the optimal algorithm is a single-copy algorithm, and the cost of any sub-optimal single-copy algorithm is not greater than twice that of the optimal single-copy algorithm.

Second, the constraints on the volume of communications and the number of copies are studied in the form of upper bounds that can be used during a scheduling interval. the upper bounds are given either on per-item basis or on all-item basis, which make it flexible enough to describe the restrictions. It is optimize the total cost under these constraints by proposing a suite of optimal solutions in polynomial time given that the upper bound is polynomially bounded by the number of requests and the number of distinct data items. It is validated the algorithms by implementing a data staging solver whereby conducting extensive simulation studies.

As for the case with a heterogeneous cost model, it is generally believed that this problem is NP-Complete. It is considered its optimality in a very restricted yet practical form which is tractable when no simultaneous communications are allowed and the maximum of the instant copies is upper bounded by a small constant. For
more general case without restriction, it is proposed a simple greedy algorithm and show its staging cost is at most twice of the optimum.

**Problem Definition**

Data staging and caching operations are handled with cost factors. Data staging process handles the process of data placement under cloud sites. Data caching is performed to improve the data transmission process. Single copy and multiple copy models are used to manage the transmission and caching cost. Upper bound is assigned to control cost under multiple copy based model. Number of service requests and the number of distinct data items are used in the upper bound estimation process. Homogeneous and heterogeneous cost models are used in the data staging process. Optimal data staging algorithm and greedy algorithm are used in the system. The following drawbacks are identified from the system.

- Multiple copy based data caching is not supported
- Bulk data transfer based cost assignment is not provided
- Multiple copy based cache management is not available
- Dynamic cache cost estimation is not provided

**Proposed System**

Under the homogeneous cost model results include two parts. First, when there are no resource constraints, it gives a DP-based optimal algorithm in polynomial time to stage and cache $k$ distinct items to satisfy a pre-determined sequence of requests, each item having single or multiple copies to minimize the cost.

Second, the constraints on the volume of communications and the number of copies are studied in the form of upper bounds that can be used during a scheduling interval. The upper bounds are given either on peritem basis or on all-item basis, which make it flexible enough to describe the restrictions. It optimizes the total cost under these constraints by proposing a suite of optimal solutions in polynomial time given that the upper bound is polynomially bounded by the number of requests and the number of distinct data items. It validates the algorithms by implementing a data staging solver whereby conducting extensive simulation studies.

Heterogeneous cost model, it is generally believed that this problem is NP-Complete. It thus consider its optimality in a very restricted yet practical form which is tractable when no simultaneous communications are allowed and the
maximum of the instant copies is upper bounded by a small constant. For more general case without restriction, it proposed a simple greedy algorithm and show its staging cost is at most twice of the optimum.

Advantages of the Proposed System

The data staging schemes are used to place the frequently used data values in cloud storages. The data values are updated in item based manner. The system supports homogeneous and heterogeneous cost models. The system also support single and multiple copy based models. Single and multiple item based data transfer process are supported by the system.

- The system minimizes the monitory cost for data caching process
- The system supports homogeneous and heterogeneous cost models
- Single and multiple copy based cache management model
- The system reduces the data transmission delay

CONCLUSION

Cloud Service Providers (CSP) shares the data under cloud data servers. Data staging techniques are used to locate the data with the consideration of time and access patterns. Data caching schemes are enhanced to manage single and multiple copies of data. The caching scheme is also enhanced with item based transmission models. Item based cache assignment scheme is supported. User access and data caching are associated with cost factors. The system improves the data transmission rate. Efficient cache space management.

REFERENCES


