Data Transfer Scheduling for Maximizing Throughput of Big-Data Computing in Cloud Systems

ABSTRACT

- Many big-data computing applications have been deployed in cloud platforms. These applications normally demand concurrent data transfers among computing nodes for parallel processing.
- It is important to find the best transfer scheduling leading to the least data retrieval time – the maximum throughput in other words.
- However, the existing methods cannot achieve this, because they ignore link bandwidths and the diversity of data replicas and paths.

CONTINUE

- In this paper, we aim to develop a max-throughput data transfer scheduling to minimize the data retrieval time of applications.
- Specifically, the problem is formulated into mixed integer programming, and an approximation algorithm is proposed, with its approximation ratio analyzed.

EXISTING SYSTEM

- Many big-data computing applications have been deployed in cloud platforms. These applications normally demand concurrent data transfers among computing nodes for parallel processing.
- It is important to find the best transfer scheduling leading to the least data retrieval time – the maximum throughput in other words.
- However, the existing methods cannot achieve this, because they ignore link bandwidths and the diversity of data replicas and paths.

PROPOSED SYSTEM

- In this paper, we aim to develop a max-throughput data transfer scheduling to minimize the data retrieval time of applications.
- Specifically, the problem is formulated into mixed integer programming, and an approximation algorithm is proposed, with its approximation ratio analyzed.
- The extensive simulations demonstrate that our algorithm can obtain near optimal solutions.

CONTINUE

- To minimize the data retrieval time (*i.e.*, to maximize the throughput) of an application consisting of concurrent tasks, we propose a max-throughput data transfer scheduling, utilizing both replica and path diversities.
- In our method, the problem is formulated into mixed integer programming, and an approximation algorithm is proposed, with its approximation ratio analyzed.
- We also solve the data retrieval problem for the case of multiple applications.

HARDWARE REQUIREMENTS

Processor	-	Pentium –III
Speed	-	1.1 Ghz
RAM	-	256 MB(min)
Hard Disk	-	20 GB
Floppy Drive	-	1.44 MB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

SOFTWARE REQUIREMENTS

Operating System	:	Windows 8
Front End	:	Java /DOTNET
Database	:	Mysql/HEIDISQL

CONCLUSION

- In this paper, we investigate the data retrieval problem in the DCN, that is to jointly select data replicas and paths for concurrent data transfers such that data retrieval time is minimized (*i.e.*, throughput is maximized).
- We propose an approximation algorithm to solve the problem, with an approximation ratio of RL, where R is the replication factor of data and L is the largest number of candidate paths.

CONTINUE

- We also solve the data retrieval problem for the case of multiple applications, keeping fairness among them.

REFERENCE

[1] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[2] D. Borthakur, “HDFS architecture guide,” *Hadoop Apache Project*, p. 53, 2008.

[3] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, “BCube: A high performance, server-centric network architecture for modular data centers,” in *SIGCOMM*, 2009, pp. 63–74.

CONTINUE

[4] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, “VL2: A scalable and flexible data center network,” in SIGCOMM, 2009, pp. 51–62.

[5] C. Hopps, “Analysis of an equal-cost multi-path algorithm,” RFC 2992, IETF, 2000.

[6] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic flow scheduling for data center networks,” in NSDI, 2010, pp. 19–19..