Resource Choice in Large Level Scattered Systems By Means of Accessibility of Information

Bachina Anusha #1,T.V.Sai Krishna #2
M.Tech Student, Associate Professor
Qis College of Engg& Tech, Ongole, A.P.

Abstract-*Scientific applications are data intensive and require access to a significant amount of dispersed data. Hence, in order to accommodate data-intensive applications in loosely coupled distributed systems, it is essential to consider not only the computational capability, but also the data accessibility of computational nodes to the required data objects. We introduce the notion of accessibility to capture both availability and performance. An increasing number of data-intensive applications require not only considerations of node computation power but also accessibility for adequate job allocations. For instance, selecting a node with intolerably slow connections can offset any benefit to running on a fast node. In this project, we present accessibility-aware resource selection techniques by which it is possible to choose nodes that will have efficient data access to remote data sources. We show that the local data access observations collected from a node's neighbors are sufficient to characterize accessibility for that node. The suggested techniques are also shown to be stable even under churn despite the loss of prior observations.*

Keywords—*Data Accessibility, resource choice, large-level scattered systems*

1. Introduction:

LARGE-SCALE distributed systems provide an attractive scalable infrastructure for network applications. This virtue has led to the deployment of several distributed systems in large-scale, loosely coupled environments such as peer-to-peer computing, distributed storage systems and Grids. In particular, the ability of large-scale systems to harvest idle cycles of geographically distributed nodes has led to a growing interest in cycle sharing systems and home projects.. In such an environment, even a few megabytes of data transfer between poorly connected nodes can have a large impact on the overall application performance. This has severely restricted the amount of data used in such computation platforms, with most computations taking place on small data objects. Emerging scientific applications, however, are data intensive and require access to a significant amount of dispersed data. Such data-intensive applications encompass a variety of domains such as high-energy physics, climate prediction, astronomy, and bioinformatics. Data availability has been widely studied over the past few years as a key metric for storage systems. However, availability is primarily used as a server-side metric that ignores client-side

accessibility of data. While availability implies that at least one instance of the data is present in the system at any given time, it does not imply that the data are always accessible from any part of the system. Similarly, the availability metric is silent about the efficiency of access from different parts of the network. Therefore, in the context of data-intensive applications, it is important to consider the metric of data accessibility: how efficiently a node can access a given data object in the system. The challenge we address is the characterization of accessibility from individual client nodes in large distributed systems. This is complicated by the dynamics of wide-area networks, which rule out static a priori measurement, and the cost of on-demand information gathering, which rules out active probing. Additionally, relying on global knowledge obstructs scalability, so any practical approach must rely on local information. To achieve accessibility-aware resource selection, we exploit local historical data access observations. This has several benefits. First, it is fully scalable as it

does not require global knowledge of the system. Second, it is inexpensive as we employ observations of the node itself and its directly connected neighbors (i.e., one-hop away). Third, past observations are helpful to characterize the access behavior of the node. We infer the latency to the server based on the prior neighbor measurement without explicitly probing the server. For this, we extend existing estimation heuristics to more accurately work with a limited number of neighbors. Our enhancement gives accurate results even with only a few neighbors. . We present accessibility-aware resource selection techniques based on our estimation functions and compare to the optimal and conventional techniques such as latency-based and random selection. Our results indicate that our approach not only outperforms the conventional techniques, but does so over a wide range of operating conditions.

In particular, the ability of large-scale systems to harvest idle cycles of geographically distributed nodes has led to a growing interest in cycle sharing systems and home projects.

But a major challenge in these systems is the network unpredictability and limited bandwidth available for data dissemination. It is expensive and it is not scalable.

The Proposed System model l consists of a network of compute nodes that provide computational resources for executing application jobs and data nodes that store data objects required for computation. In our context, data objects can be files, database records, or any other data representations.

In this system we assume that both compute and data nodes are connected in an overlay structure. We do not assume any specific type of organization for the overlay. It can be constructed by using typical overlay network architectures such as unstructured and structured or any other techniques.

We assume that the system provides built-in functions for object store and retrieval so that objects can be disseminated and accessed by any node across the system. Each node in the network can be a compute node, data node, or both.

In this System accessibility is to capture both availability and performance.

The system provides built-in functions for object store and retrieval so that objects can be disseminated and accessed by any node across the system.

It is inexpensive and scalable when compare to the existing system.

This is used to develop a system with minimum number of neighbors and accessibility-based resource selection.

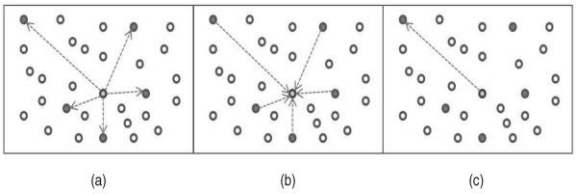


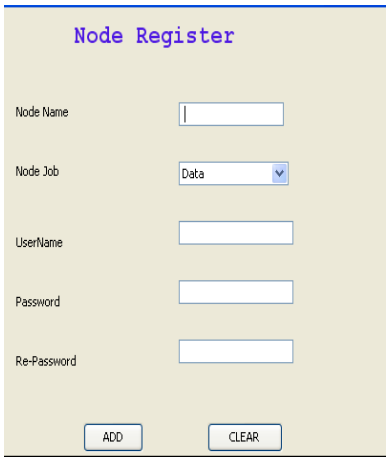
Fig. 1. Accessibility-based resource selection. (a) Initiator asks accessibility estimation to candidates. (b) Candidate offers estimated accessibility to initiator. (c) Initiator selects the best candidate based on the reported accessibility.

2. MODULES:

- Designing The System Model
- Checking Accessibility
- Allocate Job to the Best node.

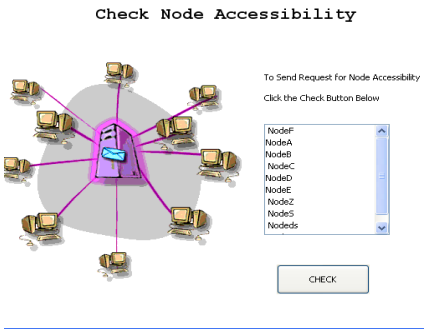
2.1 Designing the System Model

Our system model consists of a network of compute nodes that provide computational resources for executing application jobs and data nodes that store data objects required for computation. In our context, data objects can be files, or any other data representations. We assume that both compute and data nodes are connected in an overlay structure. We do not assume any specific type of organization for the overlay. It can be constructed by using typical overlay network architectures such as unstructured and structured or any other techniques. However, we assume that the system provides built-in functions for object store and retrieval so that objects can be disseminated and accessed by any node across the system. Each node in the network can be a compute node, data node, or both.



2.2 Checking Accessibility

The Initiator node asks accessibility estimation to candidate nodes. The Candidate offers estimated accessibility to initiator. An initiator selects a compute node from a set of candidates.



2.3 Allocate Job to the Best node.

Once the initiator selects a node, the job is transferred to the selected node, called a worker. The worker then downloads the data object required for the job from the network and performs the computation. When the job execution is finished, the worker returns the result to the initiator.

3. Technique Used/Algorithm Used:

Accessibility-Based Resource Choice:

Job Ji is defined as a computation unit which requires object oi to complete the task. We assume that objects, e.g., oi, have already been staged in the network and perhaps replicated to a set of nodes Ri ¼ fr1i ; r2i ; :::g based upon projected demand.

The job Ji is submitted by the initiator. From the given candidates C ¼ fc1; c2; :::g, the initiator selects one (i.e., worker 2 C) to allocate the job.

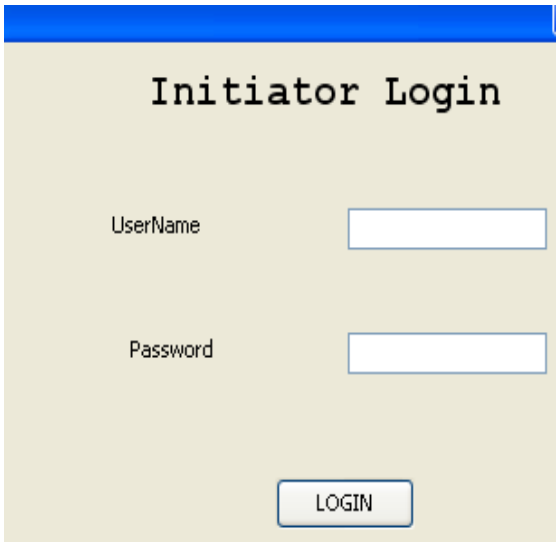
The selection heuristic Hs is defined as follows:

Hs : C → cm such that

Accessibility $(J_i) = \min_{n=1---,|C|}$ accessibility $_{cm}(J_i)$

4. Input Design:

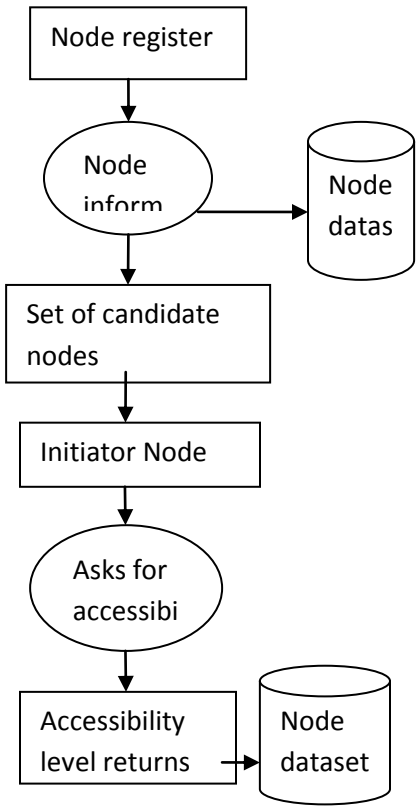
The neighbor node should login and register in the network .The initiator should login with user name and password. The initiator send request to all neighbor node for accessibility level .



5. Output Design:

The Initiator select the best node and assign job to the node the best node returns the completed job to the initiator.

DATA FLOW DIAGRAM



Advantages:

This Project present decentralized, scalable, and efficient resource selection

Our techniques rely only on local, historic observations, so it is possible to keep network overhead tolerable.

Our techniques outperform conventional approaches and are reasonably close to the optimal selection.

6. Application:

Data-intensive applications in loosely coupled distributed systems. Such applications require more sophisticated resource selection due to bandwidth and connectivity unpredictability.

7. Future Work:

The Project is focused on performance for the accessibility metric. The next step is to capture availability as well as performance to take dynamism into account. In addition to this, we plan to extend our work by providing system-wide dissemination of observations so that the node has more chances to see relevant observations in estimation. This is reasonable since neighbor estimation has no constraints on topological or geographical similarities to Utilize observations coming from other nodes.

8. Conclusion

Accessibility is a crucial concern for an increasing number of data-intensive applications in loosely coupled distributed systems. Such applications require more sophisticated resource selection due to bandwidth and connectivity unpredictability. In this project, we presented decentralized, scalable, and efficient resource selection techniques based on accessibility. our estimation techniques are sufficiently accurate to provide a meaningful rank order of nodes based on their accessibility.

REFERENCES

[1] D.P. Anderson and G. Fedak, "The Computational and Storage

[2] A. Haeberlen, A. Mislove, and P. Druschel, "Glacier: Highly Durable, Decentralized Storage Despite Massive Correlated Failures,"Proc. Symp. Networked Systems Design and Implementation(NSDI '05), May 2005.

[3] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi,S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao,"Oceanstore: An Architecture for Global-Scale Persistent Storage,"Proc. ACM Int'l Conf. Architectural Support for ProgrammingLanguages and Operating Systems (ASPLOS '07), Nov. 2000.KIM ET AL.: USING DATA ACCESSIBILITY FOR RESOURCE SELECTION IN LARGE-SCALE DISTRIBUTED SYSTEMS 799

[5] A. Chien, B. Calder, S. Elbert, and K. Bhatia, "Entropia: Architecture and Performance of an Enterprise Desktop Grid System," J. Parallel and Distributed Computing, 49] D.P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "Seti@home: An Experiment in Public-Resource Computing," Comm. ACM, vol. 45, no. 11, pp. 56-61, 2002.

[6] "Search for Extraterrestrial Intelligence (SETI) Project,"

[7] "BOINC: Berkeley Open Infrastructure for Network Computing,"

[8] N. Massey, T. Aina, M. Allen, C. Christensen, D. Frame, D. Goodman, J. Kettleborough, A. Martin, S. Pascoe, and D. Stainforth,"Data Access and Analysis with Distributed Federated DataServers in climateprediction.net," Advances in Geosciences, vol. 8,pp. 49-56, June 2006.

[9] G.B. Berriman, A.C. Laity, J.C. Good, J.C. Jacob, D.S. Katz, E.Deelman, G. Singh, M.-H. Su, and T.A. Prince, "Montage: The Architecture and Scientific Applications of a National Virtual Observatory Service for Computing Astronomical Image Mosaics,"

[10] "BLAST: The Basic Local Alignment Search Tool," http:// www.ncbi.nlm.nih.gov/blast, 2009.

[11] W. Hoschek, F.J. Jae´n-Martı´nez, A. Samar, H. Stockinger, and K. Stockinger, "Data Management in an International Data Grid Project," Proc. IEEE/ACM Int'l Conf. Grid Computing (GRID '00), pp. 77-90, 2000.

[12] Y.-M. Teo, X. Wang, and Y.-K. Ng, "Glad: A System for Developing and Deploying Large-Scale Bioinformatics Grid," Bioinformatics, vol. 21, no. 6, pp. 794-802, 2005.

[13] S. Hotz, "Routing Information Organization to Support Scalable Interdomain Routing with Heterogeneous Path Requirements,"PhD dissertation, 1994.

[14] J.D. Guyton and M.F. Schwartz, "Locating Nearby Copies of Replicated Internet Servers," SIGCOMM Computer Comm. Rev., vol. 25, no. 4, pp. 288-298, 1995.

[15] E. Ng and H. Zhang, "Predicting Internet Network Distance with Coordiantes-Based Approaches," Proc. IEEE INFOCOM

[16] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks," Proc. ACM SIGCOMM

[17] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," Proc. ACM SIGMETRICS '02, pp. 258-259, 2002.

[18] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker,"A Scalable Content-Addressable Network," Proc. ACM SIGCOMM

[19] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems,"

[20] "PlanetLab Iperf," http://jabber.services.planet-lab.org/php/800 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 6, JUNE 2009 Jinoh Kim received the BE and MS degrees in computer science and engineering from Inha University, Korea, in 1991 and 1994, respectively. From 1991 to 2005, he was with ETRI, Korea, where he worked on ATM management, IP over ATM, network security, and policy-based security management. His research interests are distributed computing including peer-to-peer computing and high-performance computing, distributed systems, and network security and management. He is a student member of the IEEE and the IEEE Computer Society. Abhishek Chandra received the BTech degree in computer science and engineering from IIT Kanpur, India, in 1997, and the MS and PhD degrees in computer science from the University of Massachusetts Amherst in 2000 and 2005, respectively. He is currently an assistant professor in the Department of Computer Science and Engineering at the University of Minnesota. His research interests are in the areas of operating systems, distributed systems, and computer networks. He received the US National Science Foundation CAREER award in 2007, and his dissertation titled "Resource Allocation for Self-Managing Servers" was nominated for the ACM Dissertation Award in 2005. He is a member of the ACM, the IEEE, the IEEE Computer Society, and USENIX.