Russ Rew*
Unidata Program Center, Boulder, Colorado

1. INTRODUCTION

New algorithms, protocols, and middleware for peer-to-peer applications are currently active subjects for computer science research. Recent advances in this area have applications to data management and data access in the geosciences. Peer-to-peer approaches may offer superior alternatives to client-server or Grid architectures for some kinds of distributed systems involving data discovery, data access, and content distribution.

This paper describes a few potential geoscience applications for recently developed peer-to-peer technologies for sharing distributed resources. By looking at peer-to-peer approaches, we hope to provoke interest and serious consideration for applications to future scientific data distribution and data access infrastructures.

2. WHAT IS P2P?

Peer to peer (P2P) systems use decentralized approaches to sharing distributed resources (e.g. computing, storage, or communication), avoiding storing or maintaining global state. Typically, P2P systems are designed to adapt to nodes entering and leaving the network and to balance resource use among member nodes. These characteristics tend to make P2P systems highly scalable.

For example, an early P2P application, SETI@Home [Anderson 2002], has used over 4 million computers to perform over 3.8×10^{21} floating point operations. P2P file sharing systems make petabytes of storage available for sharing by millions of users. We contend that some of the mechanisms, algorithms, and protocols that have been developed to support P2P systems may be useful applications involving only hundreds of nodes that are common in the geosciences today, and that scaling these applications for future uses may require P2P approaches.

Below we consider some example problems, and in each case we briefly describe a conventional approach and a P2P approach. Potential benefits of the P2P approach are presented, but without giving a carefully balanced consideration of current practicality, because often the P2P approaches are too new to have undergone large-scale testing or refinement. Nevertheless, we hope to interest some curious readers to seek more complete descriptions and learn more about the novel ideas and benefits of current P2P research in the references provided.

*Corresponding author address: Russ Rew, Unidata/UCAR, PO Box 3000, Boulder, CO 80307;e-mail russ@unidata.ucar.edu. Unidata is sponsored by the National Science Foundation.

3. LOCATION-INDEPENDENT DATA ACCESS

The idea of accessing data by its characteristics rather than by where it is stored is an attractive one. One traditional approach requires the creation and maintenance of indexes to datasets on catalog servers. Data is accessed by indirection through the catalog server by using metadata to look up one or more URLs where the data may be found. Applications use the catalog servers to access desired data without specifying its location on the network. Location-transparency is achieved by indirection. Drawbacks of this approach include the difficulty of keeping centralized catalogs up-to-date with continuously growing data collections, single points of failure, and bottlenecks to scalability when many clients need to use lookup services simultaneously.

A P2P approach to this problem makes use a Distributed Hash Table (DHT) infrastructure to store data objects and replicas in a network of hosts and makes them accessible by key rather than by location. For geoscience applications, the key could be derived from a canonical representation of the metadata known to both producers and consumers of the data. In DHTs, access to the data by key is distributed, using recently developed protocols on structured network overlays to route through *n* hosts using only on the order of *log(n)* messages to locate the data. Encryption techniques protect against inadvertent or malicious modification of the stored data. An important characteristic of these P2P solutions is their automatic preservation of efficient access guarantees as new data is added to the collection and as new hosts enter or current hosts leave the network. Existing examples of DHT implementations are built on key-based routing systems such as CAN [Ratnasamy 2001], Chord [Stoica 2001], Viceroy [Malkhi 2002], and Pastry [Rowstron 2001].

The benefits of the DHT approach are resilience to node failures, scalability, and location independence using only local resources. Location of data objects in a DHT can succeed in the face of dynamic network changes and widespread failures, as a result of the design for resilience.

4. APPLICATION-LEVEL MULTICAST

Consider the problem of distributing near-real time data to a set of subscribing sites in a way that

delivers the data very soon after it is injected into the network, perhaps from multiple source sites. The general problem is how to implement multicast efficiently, with a few producers of data and many consumers of the data on a network.

One traditional approach uses so-called reliable multicast implemented over IP multicast. However, IP multicast is not widely deployed for various reasons, including well-known scalability problems. Scalability is impractical with many multicast groups, because each router must maintain state for every multicast group that makes use of that router.

Another conventional approach establishes a tailored distribution network overlay, as exemplified by Unidata's LDM [Unidata 2003]. This client/server system supports the Internet Data Distribution system, serving hundreds of sites with near real-time atmospheric science and related data. A static routing topology for each data stream is constructed and maintained, with data relayed using unicast from host to host in the routing tree to get from source sites to destination sites. A problem with this approach is scalability: manually maintaining static routing for many data streams with changing memberships is resource-intensive, and results in suboptimal routing topologies. Another problem is that a relatively small number of relay nodes carry most of the load of forwarding data to a large number of leaf nodes.

A P2P approach supports scalability to a much larger number of sites, maintaining the multicast routing tree automatically through the use of self-organizing protocols that adapt to changes in the underlying network and that permit hosts to enter and leave the network without disrupting the data flows. A recent example of P2P research supporting this application is SplitStream [Castro 2003], which implements a clever idea for distributing and sharing the forwarding load in a multicast system among all the participants. Remarkably, SplitStream P2P multicast can deliver data efficiently, even if each node contributes only as much forwarding bandwidth as it receives, providing an elegant solution to the problem of overloaded relay nodes.

Another promising P2P approach is exemplified by Bullet [Kostic 2003], in which a distribution mesh is used instead of a distribution tree, and nodes simultaneously receive a desired data object from multiple sources in parallel, delivering fundamentally higher bandwidth throughput than is possible with a distribution tree.

Any P2P system that supports key-based routing on a structured overlay network (which includes systems based on CAN, Chord, Pastry, Tapestry [Zhao 2003], and Viceroy) can be shown to also support efficient application-level multicast, as well as *anycast* where the goal is to send a message to any one member of a communication group.

5. COOPERATIVE ARCHIVAL STORAGE

An important problem in the geosciences is archiving large and growing collections of data reliably, so that they may be preserved for search and access by future researchers and applications.

The conventional client-server solutions establish centralized archive server sites, perhaps replicated for availability and redundancy, with data backed up to alternate media. When new data is added to the archive, metadata is derived or constructed and added to catalogs or indexes. All or parts of large datasets are accessed using standard clients or through protocols such as FTP, HTTP, or OPeNDAP. Durability is achieved through replication, backup, and careful stewardship of the data, reading and copying it to new media periodically.

A decentralized P2P approach would instead store the same data archives across many autonomous host computers that occasionally enter and leave the P2P network overlay. To keep the data available, sufficient multiple copies or encoded fragments are stored, so that there is an extremely high probability of the availability of any desired data or of the encoded fragments from which it can be assembled whenever access is desired. The system can be designed to handle redundant storage automatically and more efficiently than replication, using appropriate encodings.

One such P2P network storage system is CFS, the Cooperative File System, [Dabek 2001]. CFS provides guarantees of efficiency, robustness, and load balance with a completely decentralized, scalable, and secure architecture. It presents stored data to applications using an ordinary file system interface, but implements this using a peer-to-peer Chord lookup service for storage blocks.

A considerably more ambitious example of a P2P approach to providing archival storage is OceanStore, a visionary proposal for a Global persistent storage infrastructure [Kubiatowicz 2000, Rhea 2001]. OceanStore, a global persistent data store designed to scale to billions of users, is based on a utility model that uses thousands or millions of servers to provide a persistent data storage service to clients. It assumes that some subset of its components will be failing at any time, and is thus designed to be self-maintaining, recovering from server and network failures, incorporating new resources, and adjusting to changing usage patterns without manual intervention. OceanStore makes use of Tapestry's decentralized object location and routing infrastructure and *erasure encoding*, a sophisticated way to store data durably that has significant advantages over replication. For example, see [Weatherspoon 2002], where a self-repairing, resilient distributed storage infrastructure using erasure codes is compared to a similar system using replication, and shown to have a mean time to failures that is many

orders of magnitude higher than replication, using similar storage and bandwidth.

Pond [Rhea 2003] is a recently developed prototype of OceanStore that demonstrates many of its features, including location-independent routing, Byzantine update commitment, continuous archiving to erasure-coded form, and use of self-maintaining algorithms. In a wide area network, Pond outperforms NFS in read-intensive applications. As a working subset of OceanStore, it points out where research is still needed to fully implement the vision.

6. SHARING COMPUTING RESOURCES

A popular approach to sharing computing resources is to structure the resources into a Grid computing utility, using client/server protocols and Web services as are provided, for example, with the Globus Toolkit [Foster 2002].

The P2P approach is exemplified by SETI@Home and a similar project for Climate modeling research, climateprediction.net, [Stainforth 2002, Allen 2003]. The climateprediction.net experiment runs a climate prediction model using idle capacity on home, school, and work computers. As of November 2003, climateprediction.net had run over 500,000 climate model years and was using over 20,000 machines, with over 5,000 runs completed. By running the model thousands of times as a large ensemble the researchers hope to find out how the model responds to small changes in its approximations, and thus evaluate model sensitivities without using supercomputers. It is designed to meet the goal of improving methods to quantify the uncertainties in climate model forecasts.

7. OTHER APPLICATIONS

Distributed data discovery requires developing ways for users or applications to find network-accessible data matching specified queries. Conventional solutions include client-server solutions that lookup data in metadata databases using various query interfaces. An example of a recent P2P solution is Edutella [Nejdl 2002], which supports distributed querying of RDF metadata in databases. An overview of other P2P approaches is available in [Joseph 2003].

Other applications for which P2P approaches appear promising include an event notification service [Cabrera 2001], and sharing resources to download large files [Maymounkov 2003].

8. SUMMARY

According to [Soto 2003], applications best suited for P2P implementation are those where

- Centralization is not possible or desired
- Massive scalability is desired
- Relationships are transient or ad-hoc
- Resources are highly distributed
- Resilience is desired

Many Earth science applications have one or more of these characteristics, so may be good candidates for a P2P approach. In particular, we believe recent P2P research has important applications for data distribution, scalable data access, and data persistence. Self-organizing data communities may benefit from the high fault tolerance and global scalability of solutions developed for P2P applications. Self-adaptive protocols, erasure encoding, and Distributed Hash Tables are just some of the techniques used in P2P systems that have wider applications.

Some of the problems P2P technologies address (for example, hundreds of thousands of applications needing to download the same file concurrently) may far exceed the current scalability requirements in the geosciences, but that could change: consider the phenomenon of "flash crowds" trying to access hurricane land fall forecasts and data associated with other severe weather events.

In planning future infrastructure for the geosciences, P2P technologies deserve to be considered for their benefits in scalability, capacity, resilience, availability, and self-organizing properties.

REFERENCES:

Allen, M., 2003: Possible or probable? *Nature*, **425**, 18 242. http://www.climateprediction.net/science/pubs/nature_18_9_03.pdf

Anderson D., J. Cobb, E. Korpela, M. Lebofsky, D. Werthimer, 2002: SETI@home: An Experiment in Public-Resource Computing. *Communications of the ACM*, **45**, 11 56-61. http://setiathome.ssl.berkeley.edu/cacm/cacm.html

Cabrera, L., M. Jones, and M. Theimer, 2001: Herald: Achieving a global event notification service. *Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems*, Elmau/Oberbayern, Germany. http://research.microsoft.com/research/sn/Herald/papers/HotOS8/HotOS8.html

Castro M., P. Druschel, A. Kermarrec, A. Nandi, A. I. T. Rowstron, A. Singh, 2003: SplitStream: high-bandwidth multicast in cooperative environments. *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, 298-313. http://www.cs.rochester.edu/sosp2003/papers/p159-castro.pdf

Dabek, F, M. Kaashoek, D. Karger, R. Morris, and I. Stoica, 2001: Wide Area Cooperative Storage

with CFS. *Proceedings of the 18th ACM Symposium on Operating Systems Principles.* http://www.pdos.lcs.mit.edu/papers/cfs:sosp01/cfs_sosp.pdf

Foster, I., C. Kesselman, J. Nick, and S. Tuecke, 2002: Grid Services for Distributed System Integration. *Computer*, **35**(6). http://www.globus.org/research/papers/ieee-cs-2.pdf

Joseph, S. and T. Hoshiai, 2003: Decentralized Meta-Data Strategies: Effective Peer-to-Peer Search. *IEICE Trans. Commun.*, **E86-B**, 6. http://search.ieice.org/2003/pdf/e86-b_6_1740.pdf

Kostic D., A. Rodriguez, J. Albrecht, A. Vahdat, 2003: Bullet: high bandwidth data dissemination using an overlay mesh. *Proceedings of the 19th ACM Symposium on Operating System Principles*, 282-297. http://www.cs.duke.edu/~vahdat/ps/bullet-sosp03.pdf

Malkhi D., M. Naor, and D. Ratajczak, 2002: Viceroy: A scalable and dynamic emulation of the butterfly. *Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing (PODC).* http://www.wisdom.weizmann.ac.il/~naor/viceroy.pdf

Maymounkov, P. and D, Mazieres, 2003: Rateless Codes and Big Downloads. *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03),* Berkeley, CA. http://iptps03.cs.berkeley.edu/final-papers/rateless_codes.ps

Nejdl W., B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, T. Risch, 2002: EDUTELLA: A P2P Networking Infrastructure Based on RDF. *The Eleventh International World Wide Web Conference*, Honolulu, Hawaii, http://edutella.jxta.org/reports/edutella-whitepaper.pdf

Ratnasamy, S., P. Francis, M. Handley, R. Karp, and S. Schenker, 2001: A scalable content-addressable network. *Proceedings of SIGCOMM 2001,* Association for Computing Machinery. http://www.acm.org/sigs/sigcomm/sigcomm2001/p13-ratnasamy.pdf

Rhea, S., C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, 2001: Maintenance-Free Global Data Storage, *IEEE Internet Computing,* **5**(5), 40-49. http://oceanstore.cs.berkeley.edu/publications/papers/pdf/ieeeic.pdf

Rhea S., P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz, 2003: Pond: the OceanStore Prototype. *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST '03).* http://oceanstore.cs.berkeley.edu/publications/papers/pdf/fast2003-pond.pdf

Rowstron A. and P. Druschel, 2001: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *Proceedings of Middleware 2001.* http://research.microsoft.com/~antr/PAST/pastry.pdf

Stainforth D., J. Kettleborough, A. Martin, A. Simpson, R. Gillis, A. Akkas, R. Gault, M. Collins, D. Gavaghan, and M. Allen, 2002: Climateprediction.net: design principles for public resource modeling research, *Proc. 14th IASTED conference on parallel and distributed computing systems* (PDCS 2002), Cambridge, MA. http://www-atm.physics.ox.ac.uk/user/das/pubs/iasted_final.pdf

Stoica I., R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, 2001: Chord: A scalable peer-to-peer lookup service for Internet applications. *Proceedings of SIGCOMM 2001*, Association for Computing Machinery. http://pdos.lcs.mit.edu/papers/chord:sigcomm01/chord_sigcomm.pdf

Stoica, I., D. Adkins, S. Ratnasamy, S. Shenker, S. Surana and Shell Zhuang, 2002: Internet Indirection Infrastructure. *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS 2002),* Cambridge, MA. http://www.cs.rice.edu/Conferences/IPTPS02/166.pdf

Soto, J. C., 2003: Introduction to Project JXTA. *Internet 2, P2P Working Group*, Indianapolis, IN. http://www.internet2.edu/presentations/fall-03/20031014-P2P-Soto.pdf

Unidata Local Data Manager (LDM) web site, 2003: http://my.unidata.ucar.edu/content/software/ldm/index.html

Weatherspoon, H. and J. Kubiatowicz, 2002: Erasure Coding vs. Replication: A Quantitative Comparison. *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS '02),* Cambridge, MA. http://oceanstore.cs.berkeley.edu/publications/papers/pdf/erasure_iptps.pdf