Are P2P Data-Dissemination Techniques Viable in Today's Data-Intensive Scientific Collaborations?

Samer Al-Kiswany[1], Matei Ripeanu[1], Adriana Iamnitchi[2] and Sudharshan Vazhkudai[3]

[1]University of British Columbia [2]University of South Florida [3]Oak Ridge National Laboratory
{samera, matei}@ece.ubc.ca anda@cse.usf.edu vazhkudaiss@ornl.gov

Abstract. The interest among a geographically distributed user base to mine massive collections of scientific data propels the need for efficient data dissemination solutions. An optimal data distribution scheme will find the delicate and often application-specific balance among conflicting success metrics such as minimizing transfer times, minimizing the impact on the network, and uniformly distributing load among participants. We use simulations to explore the performance of classes of data-distribution techniques, some of which successfully deployed in large peer-to-peer communities, in the context of today's data-centric scientific collaborations. Based on these simulations we derive several recommendations for data distribution in real-world science collaborations.

1. Introduction

Modern science is data-intensive. Large-scale simulations, new scientific instruments, and large-scale observatories generate massive volumes of data that need to be analyzed by large, geographically dispersed user communities. These trends are emerging in fields as diverse as bioinformatics and high-energy physics. Examples include the Large Hadron Collider (LHC) experiment at CERN and the DØ experiment at Fermi Lab. Aiding in the formation of these collaborative data federations are ever increasing network capabilities including high-speed optical interconnects (e.g., LambdaGrid) and highly optimized bulk transfer tools and protocols (e.g., GridFTP).

Data dissemination in such federations involves dynamic distribution of subsets of data, available at one site, to one or many collaborating locations for real-time analysis and visualization. For instance, the PetaBytes of data from the LHC experiment have to be distributed world-wide, across national and regional centers.

Two conflicting arguments compete to shape the one-to-many delivery of large scientific data over well provisioned networks. On one side, there is the intuition that the well provisioned networks are sufficient to guarantee good data-delivery performance; sophisticated algorithms that adapt to unstable or limited-resource environments are superfluous and add unjustified overheads in these environments. The counter-argument is that advanced data dissemination systems are still required as the size of data and the relatively large collaborations create contention and bottlenecks on shared resources, which hinder efficient usage. Additionally, even if contention for shared resources is not a serious concern, the question if networks are over provisioned and thus generate unnecessary costs remains.

These two arguments motivate this study: we explore, experimentally, the space of solutions for one-to-many, large-scale data delivery in today's environments via simulations. We consider solutions typically associated with peer-to-peer (P2P) applications (such as BitTorrent) and evaluate them under our target scenario of large data federations. To this end, we used both generated as well as real production Grid testbed topologies in our evaluations.

Our contribution is twofold. First, we quantitatively evaluate and compare a set of representative data-delivery techniques applied to a grid environment. The quantitative evaluation is then used to derive well-supported recommendations for choosing data-dissemination solutions and for provisioning the Grid networking infrastructure. Further, our study contributes to a better understanding of the performance tradeoffs in the data-dissemination space. To the best of our knowledge, this is the first, head-to-head comparison of alternative solutions using multiple performance metrics. Second, we propose a simulation framework that can be used to explore optimal solutions for specific deployments or can be extended for new dissemination solutions.

To derive our recommendations, we identify a relevant set of candidate solutions from different domains (Section 3), build a simulator (presented in Section 4) and evaluate the candidate solutions on key metrics such as time-to-delivery, generated overhead, and load balance (Section 5). We summarize our findings in Section 6.

2. Data in Scientific Collaborations

Today, Grids are providing an infrastructure that enables users to dynamically distribute and share massive datasets. However, most data distribution strategies currently in place involve explicit data movement through batch jobs that are seldom sympathetic to changing network conditions, congestion and latency, and rarely exploit the collaborative nature of modern-day science [1].

On the other hand, P2P file sharing and collaborative caching efficiently exploit patterns in users' data sharing behavior. However, such techniques are not directly adaptable to Grid settings because of different usage scenarios. In particular, three key differences make it difficult to predict the behavior of P2P techniques in scientific data federations: scale of data, data usage characteristics, and resource characteristics.

The *scale of data* poses unique challenges: scientific data consists of massive collections comprising of hundreds to thousands of files. For instance, of the more than one million files accessed in DØ between January 2003 and May 2005, more than 5% are larger than 1GB and the mean file size is larger than 300MB [2].

Usage of data in scientific communities varies in *intensity* compared to other communities. For example, 561 scientists from DØ processed more than 5PB of data in 29 months, which translates to accessing more than 1.13 million distinct data files [2]. However, *popularity distributions* for scientific data are more uniform than in P2P systems or in the Web. Further, in scientific environments, files are often used in groups and not individually.

Finally, *resource availability* in grids poses smaller challenges than in P2P systems. Computers stay connected for longer, with significantly lower churn rate and higher availability due to hardware and software configurations. At the same time,

data federations are overlays built atop well-provisioned networks (e.g., TeraGrid) as opposed to the commercial Internet. Additionally, *resource sharing is often enforced by out-of-band means*, such as agreements between institutions or between institutions and funding agencies. For this reason, mechanisms that enforce participation and fair sharing, such as the tit-for-tat scheme of BitTorrent, are often unnecessary.

To summarize, most Grid data distribution strategies currently in place fail to exploit the characteristics and data usage patterns emerging in today's scientific collaborations. The properties (huge data volumes, well provisioned networks, stable resources, and cooperative environments) of these data collaborations, however, invite the *question whether P2P data-distribution strategies result in tangible gains on well-endowed network infrastructures on which today's Grids are deployed.* A careful study is necessary to derive recommendations for building and provisioning future testbeds and choosing efficient dissemination approaches for science collaborations.

3. Data Distribution: Solutions and Metrics

We have identified a number of techniques as potential candidates for our comparison. We provide a classification of data distribution techniques (Section 3.1), detail the techniques we explore in this paper (Section 3.2), and present the criteria over which they are typically evaluated (Section 3.3).

3.1 Classification of Approaches

We identify three broad categories of techniques to optimize data distribution: data staging, data partitioning, and exploiting orthogonal bandwidth. In this section we describe these techniques and discuss them in the context of our target environment.

Data Staging. With data staging, participating nodes are used as intermediate storage points in the data distribution solution. Such an approach is made feasible due to the emergence of network overlays. For instance, it is becoming increasingly common practice for application-specific groups to build collaborative networks, replete with their application-level routing infrastructure. This is based on the premise that sophisticated applications are more aware of their resource needs, deadlines, and constraints and can thus perform better resource allocation and scheduling. In this vein, P2P file-sharing systems can be viewed as data-sharing overlays with sophisticated application-level routing performed atop traditional Internet.

In data grids, data staging is encouraged by the increasing significance of application-level tuning of large transfers. For instance, collaborating sites often use path information to make informed decisions to access data from preferred locations, based on a delivery constraint schedule [3]. A logical extension is thus to utilize the participating sites as intermediary data staging points for efficient data dissemination. Additionally, a data distribution infrastructure can include a set of intermediary, strategically placed resources to stage data (e.g., IBP [4]).

Data Partitioning. To add flexibility, various P2P data distribution solutions split files into blocks that are transferred independently (e.g., BitTorrent[5]). Much like the aforementioned application-level routing, this approach allows applications a greater degree of control over data distribution. Further, it enables application-level error

correction. For example, when downloading a file from multiple replicas, partitioning can be coupled with erasure coding to achieve fault tolerance. Partitioning techniques have significant value in a data grid setting. Bulk data movement in the Grid is usually long-haul transfers that have to survive a wide range of failures (e.g., network outage, security proxy expiration). Thus, there is a genuine need to provide application-level resilience for data transfers.

Orthogonal Bandwidth Exploitation. Once a basic file partitioning mechanism is in place, it can then be used to accelerate data distribution by exploiting "orthogonal bandwidth", i.e., the bandwidth that cannot be used by a traditional, source-routed data-distribution tree. This is the underlying premise in a number of commercially deployed (e.g., BitTorrent) or academically designed (e.g., Bullet [6]) data-distribution systems that owe much of their success to such optimizations.

Intuitively, it seems these techniques will have commensurate gains when applied to data grids. However, several of these optimizations are designed to work in a non-cooperative environment, where peers contend for scarce resources (e.g., bandwidth). One question to address is how this intuition translates when the bandwidth is plentiful and users are cooperative, as is the case with current scientific data collaborations.

3.2 Candidate Solutions for Evaluation

For our experimental study, we selected representative solutions from each of the categories presented above. We also include other traditional, well understood techniques for comparison. A brief description of these solutions follows (Our technical report [7] has a complete discussion).

Application-level multicast (ALM) solutions organize participating nodes into a source-rooted distribution tree overlay. What differentiates various ALM solutions is the algorithm used to build and maintain the distribution tree. For our experiments, we chose ALMI [8], a solution offering near optimal trees built using global views.

BitTorrent [5] is a popular data distribution system that exploits the upload bandwidth of participating peers for efficient data dissemination. Participating nodes build transitory pair-wise relationships and exchange missing file blocks. BitTorrent assumes a non-cooperative environment and employs a tit-for-tat incentive mechanism to discourage free riders. Each node selects its peers to minimize its own time to acquire content disregarding the overall efficiency of the data distribution operation.

Bullet [6] offers a way to exploit orthogonal bandwidth by distributing disjoint subsets of data using an initial distribution tree (we use the ALM-built tree in this study). Informed delivery techniques [9] are then used to reconcile these data sets stored at destination nodes, which exchange the necessary blocks.

Logistical multicast [4] employs strategically placed nodes in an overlay to expedite data distribution. We evaluate an idealized variation of this approach that associates storage with each router in our topologies.

Spider [10] offers a set of heuristics that enables fast content distribution by building multiple source-rooted trees (assuming global views).

3.3 Success Metrics

Multiple categories of success metrics can be defined for most data management problems. We note that the relative importance of these metrics is dependent on the

application context. Thus, no data distribution solution is optimal for all cases and a careful evaluation of various techniques is required to choose a solution appropriate for a specific application context and deployment scenario. The most representative performance objectives include:

- *Minimizing transfer times*: The application focus may be to minimize the average, median, N^{th} percentile, or the highest transfer time to destination.
- *Minimizing the overall impact on the network*: This involves minimizing the load on bottleneck links, the volume of generated traffic, or the overall network 'effort'.
- *Load balancing*: Enlisting all participating sites in the data dissemination effort makes spreading the load among them crucial.
- *Fairness*: Being fair in the presence of concurrent transfers can be an important concern depending on the lower-layer network and the protocols used.

4. Simulating Data Dissemination

To evaluate the techniques above, we have built a block-level simulator. This section presents the set of decisions that guided our simulator design. For a detailed description of the simulator we refer the reader to Al Kiswany et al. [7].

As with most simulators, the main tradeoff is between the amount of resources allocated for simulation and simulation fidelity. At one end of the possible design spectrum are packet-level simulators and emulators. These require significant hardware resources, but model application performance faithfully by running unmodified application code and simulating/emulating network transfers at the IP-packet level. At the other end of the spectrum are high-level simulators that abstract the application transfer patterns and employ only coarse network modeling. Our simulator sits in between these two extremes. From an application perspective, the granularity is file block transfer, a natural choice since many of the data dissemination schemes we investigate use file blocks as their data management unit. From a network perspective, while we do not simulate at the packet level, we do, however, simulate link level contention between application flows.

Additionally, our simulator design is guided by the following decisions:

- *Ignore control overheads.* For our target scenario, protocol control overheads are often orders of magnitude lower than the effort to transfer the actual data payload. Moreover, since the control messages overlap or are piggybacked on data transfers, the latency introduced by the control channel is often negligible.
- *Use of global views.* Our simulator uses a global view of the system in order to hide algorithmic details that are not relevant to our investigation.
- *Ignore competing traffic.* To increase simulator scalability we do not directly model competing traffic. Competing traffic can be modeled, however, by varying the available bandwidth of the links in the simulated network topology.

We experiment with the four solutions for data dissemination solutions described above: application-level multicast (ALM), BitTorrent, Bullet, and logistical multicast. To study their performance, we compare them with two base cases: IP multicast (and its improvement using Spider heuristics) and the naïve, yet popular, solution that sends separate copies of the data from the source to each destination.

For the more complex protocols, Bullet and BitTorrent, the simulator models each block transfer. This is necessary due to the non-deterministic nature of these data dissemination solutions. The simulations use a default block size of 512KB, as in deployed BitTorrent systems. We experimented with different block sizes and noticed that the block size does not significantly affect the results as long as files can be split in a large number of blocks. All our simulations explore the performance of distributing a 1GB file over different topologies.

5. Experimental Results

We use the physical network topologies of two real-world grid testbeds EGEE [11] and GridPP [12]. The EEGE topology is presented in Figure 1. Our technical report [7] presents the detailed GridPP topology. Additionally, to increase the confidence in our results, we use BRITE to generate two sets of larger (hundreds of nodes) Waxman topologies [13]. These sets have the same number of nodes and constant overall bandwidth, and differ only in the density of core links. Our goal is to compare results on these two sets of topologies to obtain a more direct measure of the degree to which various protocols are able to exploit network path diversity. Due to space limitation, we present the key observations and direct the interested reader to our detailed results, for all of these topologies, in our technical report [7].

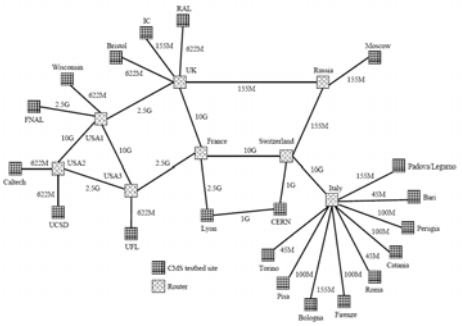

Figure 1. EGEE topology.

5.1 Performance: File Transfer Time

Depending on the application context, the performance focus can be on minimizing the average, median, N^{th} percentile, or the largest transfer time to destination. To cover all these criteria, for each data dissemination technique we present the evolution in time of the number of destinations that have completed the file transfer.

Figure 2 presents this evolution for the original EGEE topology. Note that here Spider builds only one dissemination tree and is thus equivalent to IP-multicast. In spite of the slightly different results for various topologies, the following observations are common:

- IP-multicast and Logistical Multicast are the best solutions to deliver a file to the slowest node as they optimally exploit the bandwidth on bottleneck links.
- Intermediate progress with IP-multicast is poor. The reason is that multicasting schemes do not include buffering at intermediate points in the network and limit their data distribution rate to the rate of the bottleneck link.
- Logistical Multicast is among the first to complete the file dissemination and also offers one of the best intermediate progress performance. This is partially a result of the bandwidth distribution in these two topologies: the bottlenecks are the site access links and not the links at the core of the network. As a result, Logistical Multicast is able to push the file fast through the core routers that border the final access link and thus offer near optimal distribution times.

- Application-level multicast (ALM), Bullet and BitTorrent are worse but comparable to Logistical Multicast, both in terms of finishing time as well as intermediate progress. They are able to exploit the plentiful bandwidth at the core and their performance is limited only by access link capacity of various destination nodes.
- The naïve technique of distributing the file through independent streams to each destination generally performs poorly. However, surprisingly, on these over-provisioned networks, its performance is competitive with that of other methods.

The surprisingly good performance of parallel independent transfers in these topologies clearly indicates that the network core is over-provisioned. We are interested in exploring the performance of data dissemination techniques at different core-to-access link capacity ratios for the following two reasons. First, if the core is over-provisioned, we would like to understand how much bandwidth (and eventually money) can be saved by reducing the core capacity without significantly altering the dissemination performance. Second, we aim to understand whether independent transfers perform similarly well when compared to more sophisticated techniques under different network conditions.

With these two goals in mind we ran the same simulations on a set of hypothetical topologies. These topologies are similar to the original EGEE and GridPP topologies except that the bandwidth of the core links (the links between the routers) is $^1/_2$, $^1/_4$, $^1/_8$, or $^1/_{16}$ of the original core link bandwidth.

Figure 3 presents the case of core link bandwidth equal to $^1/_8$ of the original band-

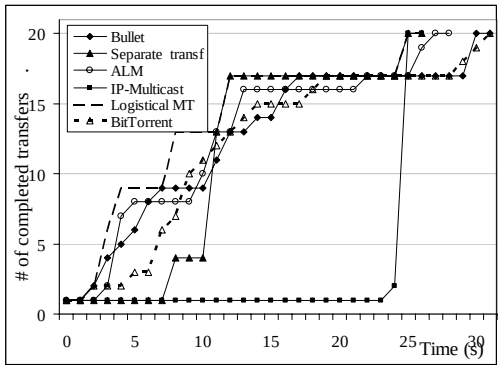

Figure 2. Number of destinations that have completed the file transfer (original EGEE topology).

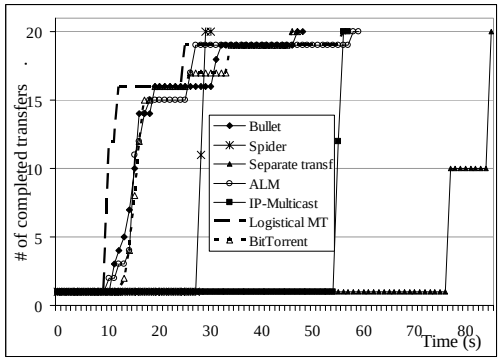

Figure 3. Number of destinations that have completed the file transfer (EGEE topology with core bandwidth reduced to $^1/_8$ of the original).

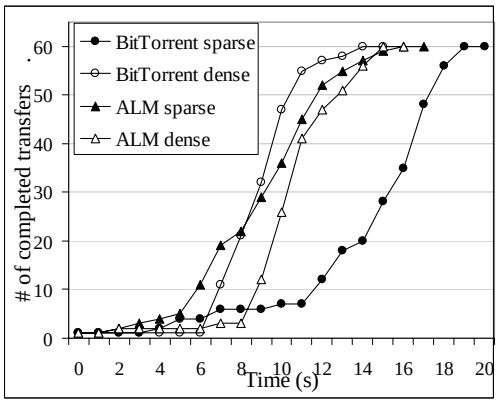

Figure 4. Number of destinations that have completed the file transfer with two generated topologies. The dense topology has four times more links in the core with 4 times less average bandwidth per link.

width in the EGEE. The performance of the parallel independent transfers degrades much faster than the performance of any other technique. Additionally, for our topologies the performance of the more sophisticated dissemination schemes does not degrade significantly when reducing the core capacity to ½ or ¼ of the original one. This is testament to their ability to exploit orthogonal bandwidth. Furthermore, it is an indication that similar performance can be obtained at lower network core budgets by employing sophisticated data distribution techniques.

To further investigate the ability to exploit alternate network paths, we generate a set of topologies in which the aggregate core bandwidth is maintained constant, but the number of core links is varied. Figure 4 compares the intermediate progress of the BitTorrent and ALM protocols on two topologies: the 'dense' topology has four times more links in the core (and four times lower link bandwidth). The results underline BitTorrent's ability to exploit all available transport capacity (Bullet shows similar behavior) unlike ALM whose relative performance degrades for denser networks.

5.2 Overheads: Network Effort

The traditional method to compare overheads for tree-based multicast solutions is to compare maximum link stress (or link stress distributions), where, link stress is defined as the number of identical logical flows that traverse the link. However, the same metric cannot be applied to Bullet or BitTorrent as these protocols dynamically adjust their data distribution patterns and, therefore, link stress varies continuously. For this reason, we propose a new metric to estimate network effort. We estimate the volume of duplicate traffic that traverses each physical link and aggregate it over all links in the testbed.

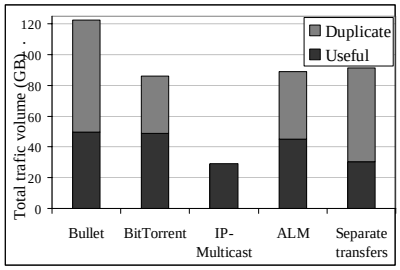

Figure 5. Overhead of each protocol on EGEE topology.

Figure 5 presents the generated useful and overhead traffic for each protocol for the original EGEE topology. We consider *useful*, the data traffic that remains after excluding all link-level duplicates. The following observations can be made from Figure 5, and can be generalized, as there is little variance across all topologies. First, as expected, IP-layer solutions do not generate any duplicates and thus are optimal in terms of total traffic. Second, Bullet, BitTorrent and ALM require significantly higher network effort. Bullet emerges as the largest bandwidth consumer. This is because it uses approximate representations of the set of blocks available at each node. False negatives on these data representations generate additional traffic overhead. BitTorrent generates slightly smaller overheads as nodes employ exact representations to denote the set of blocks available locally. Finally, ALM trees also introduce considerable overhead as the tree construction algorithm is optimized for high-bandwidth dissemination and ignores node location in the physical topology.

5.3 Load Balance

Another metric to evaluate the performance of data dissemination schemes is load balancing. To this end, we estimate the volume of data processed (both received and sent) at each end-node. Obviously, IP-layer techniques that duplicate packets at

routers or storage points inside the network, will offer ideal load balancing. Sending data through independent connections directly from the source will offer the worst load balance as the source load is proportional to the number of destinations.

Figure 6 presents the load balancing performance of the remaining techniques: ALM, BitTorrent, and Bullet. These results are obtained for the GridPP topology, but again, the relative order of these techniques in terms of load balance does not change across all our experiments.

Apart from independent transfers, ALM has the worst load balance among the three solutions as it tends to increase the load on the nodes with ample access-link bandwidth. Of the remaining two, BitTorrent offers slightly better load balancing than Bullet due to its tit-for-tat mechanism that implicitly aims to evenly spread data-dissemination efforts.

5.4 Fairness to Competing Traffic

While all the application layer protocols we analyze use TCP-friendly congestion control scheme for data exchanges between each individual pair of nodes, they differ in their impact on the network and the competing traffic. In spite of this, there is little related work on analyzing the relative fairness of data distribution schemes. We use link stress as a metric to estimate impact on competing traffic: the higher the number of flows a distribution scheme maps on a physical link, the higher the impact on competing traffic. This impact is non-negligible. The average link stress generated by Bullet can be as high as 12, while the maximum link stress can be as high as 23. This implies that, if a unicast transfer shares its bottleneck link with a link on which Bullet generates such stress, its allocated bandwidth is drastically reduced.

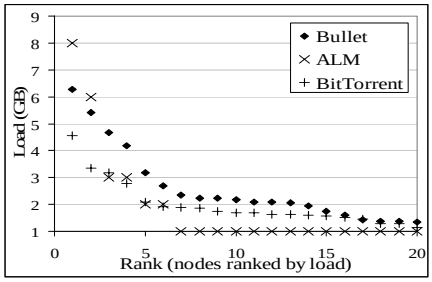

Figure 6. Load balancing for ALM, BitTorrent and Bullet. Nodes are ranked in decreasing order of their load.

6. Summary

This study focuses on the problem of one to many data dissemination in the context of today's science grids. Data dissemination in these environments is characterized by relatively small collaborations (tens to hundreds of participating sites), large data files, and well-provisioned networks. This study provides an experimentally-supported answer to the question: "Given the characteristics of deployed grids, what benefits can P2P solutions for one-to-many data dissemination offer?"

Our simulation-based comparison of seven solutions drawn from traditional data delivery systems and P2P networks shows the following:

- Some of *today*'s Grid testbeds are over-provisioned. In this case, the deployment is scalable with the size of the user community, and P2P solutions that adapt to dynamic and under-provisioned networks do not bring significant benefits. While they improve load balancing, they add significant overheads and, more importantly, do not offer significant improvements in terms of distribution time.
- Application-level schemes such as BitTorrent, Bullet and application-level multi-

cast perform best in terms of file-delivery time. However, they introduce high-traffic overheads, even higher than independent parallel transfers. On the other hand, BitTorrent and Bullet are designed to deal with dynamic environment conditions, which might be desirable in some scenarios.

- The naive solution of separate data transfers from source to each destination yields reasonable performance on well-provisioned networks but its performance drops dramatically when the available bandwidth decreases. In such cases, adaptive P2P-like techniques that are able to exploit multiple paths existing in the physical topology can offer good performance on a network that is less well provisioned.

In short, the P2P solutions that offer load balancing, adaptive data dissemination, and participation incentives, lead to unjustified costs in today's scientific data collaborations deployed on over-provisioned network cores. However, as user communities grow and these deployments scale (as already seen in Open Science Grid or TeraGrid) P2P data delivery mechanisms will outperform other techniques.

In any case, network provisioning has to progress hand-in-hand with improvements and the adoption of intelligent, adaptive data dissemination techniques. In conjunction with efficient data distribution techniques, appropriate network provisioning will not only save costs while building/provisioning collaborations, but also derive optimal performance from deployed networks.

References

1. Iamnitchi, A., Ripeanu, M., and Foster, I. *Small-World File-Sharing Communities*. in *Infocom 2004*. 2004. Hong Knog.
2. Iamnitchi, A., Doraimani, S., and Garzoglio, G. *Filecules in High-Energy Physics: Characteristics and Impact on Resource Management*. in *HPDC 2006*. 2006. France.
3. Vazhkudai, S., Tuecke, S., and Foster, I. *Replica Selection in the Globus Data Grid*. in *IEEE International Conference on Cluster Computing and the Grid (CCGRID2001)*.
4. Beck, M., Moore, T., Plank, J.S., and Swany, M. *Logistical Networking: Sharing More Than the Wires*. in *Active Middleware Services Workshop*. 2000. Norwell, MA.
5. Cohen, B., *BitTorrent web site: http://www.bittorrent.com*. 2005.
6. Kostic, D., Rodriguez, A., Albrecht, J., and Vahdat, A. *Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh*. in *SOSP'03*. 2003. Lake George, NY.
7. Al-Kiswany, S., Ripeanu, M., Iamnitchi, A., and Vazhkudai, S., *Are P2P Data-Dissemination Techniques Viable in Today's Data Intensive Scientific Collaborations?* 2007, University of British Columbia.
8. Pendarakis, D., Shi, S., Verma, D., and Waldvogel, M. *ALMI: An Application Level Multicast Infrastructure*. in *USITS'01*. 2001.
9. Byers, J., Considine, J., Mitzenmacher, M., and Rost, S. *Informed Content Delivery Across Adaptive Overlay Networks*. in *SIGCOMM2002*. 2002. Pittsburg, PA.
10. Ganguly, S., Saxena, A., Bhatnagar, S., Banerjee, S., et al. *Fast Replication in Content Distribution Overlays*. in *IEEE INFOCOM*. 2005. Miami, FL.
11. *Enabling Grids for E-sciencE Project*. 2006.
12. Britton, D., Cass, A.J., Clarke, P.E.L., Coles, J.C., et al. *GridPP: Meeting the Particle Physics Computing Challenge*. in *UK e-Science All Hands Conference*. 2005.
13. Medina, A., Lakhina, A., Matta, I., and Byers, J. *BRITE: An Approach to Universal Topology Generation*. in *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS '01*. 2001. Cincinnati, Ohio.