RC22278 (W0112-027) December 11, 2001
Computer Science

IBM Research Report

On the viability of wide-area storage services

Khalil S. Amiri, Seraphin B. Calo, Kang-Won Lee
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

On the viability of wide-area storage services

Khalil Amiri, Seraphin Calo, Kang-Won Lee

IBM T. J. Watson Research Center
{*amirik,scalo,kangwon*}*@us.ibm.com*

December 6, 2001

Abstract

Accessing storage resources over wide-area networks is enabled by recent standards such as iSCSI and FCIP which allow storage traffic to be transmitted over IP networks. Wide-area storage access is required when storage systems are hosted remotely by a storage service provider, where customer applications are remote from the "data center." Wide area networks, however, are usually characterized by high latencies and low bandwidths compared to storage-area networks. While investments in networking infrastructure combined with greater bandwidth demand may result in cost-effective bandwidths over the wide area, the relatively higher latencies of wide area communication is a problem that is likely to persist for the foreseeable future. This article investigates the viability of wide-area distance-connected storage for various application classes. Based on preliminary analysis of application workload characteristics (e.g., access patterns), we categorize key commercial applications with respect to how well they might perform in a remote storage service architecture. Secondly, we describe a customer-side storage caching appliance that alleviates the latency and bandwidth problems of wide-area storage services, and estimate its expected performance benefits using a simple analytical model.

1 Introduction

The outsourcing of storage system management promises higher availability and performance through expert-managed backup, configuration, and maintenance of storage resources. Furthermore, large initial acquisition costs are eliminated in favor of a pay-per-use model. Recently, several providers started offering management services for storage systems, including IBM Global Services (IGS), Storage Networks, Inc. (SNI), Hewlett-Packard, and EMC. Providers currently co-locate the storage systems with the servers that access them, usually either at the customer site or at a special hosting site. One potential future direction is to allow a limited number of data centers, or *storage locations* to serve a large geographically dispersed customer base.

Remote storage access is not without problems, however. Wide area networks bring two challenges: high latencies and limited bandwidths. Bandwidth over wide areas continues to be costly, currently around $125 per Mbps per month for an OC-3 link [1], or almost $ 20,000 per month for the entire link. Emerging startups,

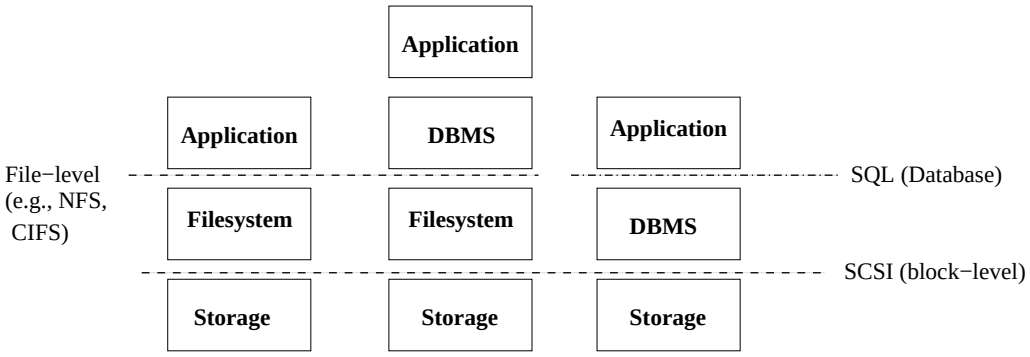

Figure 1: Typical application stacks.

however, are promising more affordable bandwidth prices in the future [4], and it is expected that investments in networking infrastructure combined with greater bandwidth demand may result in cost-effective bandwidth over the wide-area. Nevertheless, the relatively higher latencies of wide area communications are likely to persist for the foreseeable future. Fundamental limitations placed by the speed of light and technological limitations placed by the overhead of routing and switching together make the latencies over the wide area one to two orders of magnitude worse than storage area network latencies [2].

In this article, we analyze the distance implications of storage access over metropolitan-area and wide-area networks. We describe current storage architectures and hosting alternatives in Section 2. In Section 3, we estimate the performance degradation due to wide-area distance connections for various applications using a simple analytical model. Based on preliminary workload analysis, we categorize key commercial applications according to how well they might perform in a remote storage services architecture. In Section 4, we describe a caching appliance designed to alleviate the latency and bandwidth problems of wide-area access, and characterize its performance using simple queuing analysis. Finally, we draw conclusions about which application classes can and cannot achieve acceptable performance from a wide-area storage service.

2 Current storage architectures

There are two principal ways by which user applications access storage: through a filesystem or a database management system. The filesystem may be either local or distributed. Shared network file servers are used to provide storage in the case of a distributed filesystem. The shared file server can be either a general purpose workstation, or a specialized appliance [8]. While general purpose network file servers use directly attached SCSI disks for back-end storage, file server appliances use integrated (also SCSI) disks, incorporated within the appliance itself.

Most database management systems (DBMS) also rely on shared centralized database servers. While most

DBMS's rely on directly attached disk arrays for back-end storage, there is a recent and growing trend to use file server appliances to provide the back-end storage. File server appliances have enhanced storage management services such as instant snapshots and easier backup and replica management that database administrators find attractive.

Figure 1 shows the typical application stacks employed in practice. Back-end storage is provided through disks or disk-arrays. A database management system or a file system is used to provide high-level data access and management functions for applications. In particular, a database management system could be layered on top of raw disk arrays or on top of a file system. Although Figure 1 represents the logical layering of applications and data management systems, in practice the file system and storage layer may be integrated together in a single appliance, as in the case of file server appliances. One important observation is that the pervasive use of caching across the application, database and file system layers results in a good percentage of reads being captured before reaching the storage system. This is not true for writes, however, which must be propagated down to the storage system. Writes may be delayed, however, to improve performance when the application allows it. Delaying writes enables independent write requests to be coalesced or to overwrite each other resulting in a smaller number of actual writes propagated down to the storage system, an optimization known as "write canceling."

There are three alternatives to hosting storage, the first is to host block-level storage (the lowest row in Figure 1), the second is to host the file system (forcing databases at the customer to use file-level interfaces for back-end storage), and the third is to host the entire database management system. The first option implies that the database server and the file server remain at the customer site. In this case, the client to server communication is local to the customer's site, while the storage to server communication is over the wide area. The chief advantage of this solution is its simplicity and its general applicability. Note from Figure 1 that block-level storage is the lowest common denominator for all application stacks. The potential disadvantage is that some applications may move less data between client and server than they move between server and storage, making storage-only hosting perform worse than entire DBMS hosting.

Another alternative is to host both the database management system and the back-end storage system at the service provider. The advantage of this approach is that better performance can be achieved because more layers are hosted by the provider, which allows a wider range of optimizations and of service differentiation. Application bandwidth requirements at the I/O (server to storage) level weigh heavily on which hosting solution is feasible or preferable. Therefore, application I/O workload characterization is a crucial prerequisite to a feasibility study.

3 Application workload characterization

To estimate the effect of distance on application performance, it is critical to first characterize the application workload. For example, a read-intensive application with a small working set is not likely to suffer much from

having storage remote, because a small customer-side cache can absorb most of the access latencies. On the other hand, an application with 50% reads and 50% writes requiring hundreds of megabytes of bandwidth is more likely to suffer if storage is accessed over a limited-bandwidth link, since writes have to be propagated to the storage location potentially saturating the uplink bandwidth. In this section, we study a wide range of applications and characterize their I/O behavior.

	Request size (KB)	Read/write ratio	Access pattern	Capacity requirement	Throughput (IOs/s)	Bandwidth (MB/s)	Data read or written
Instructional	4 – 8KB	5.6	sequential (entire files)	$<$ GB/user	low	low	1.1GB Per hundred users per day
Research	4 – 8KB	3.7	sequential	few GB/user	low	low	3.4GB Per hundred users per day
NT office use	4 – 8KB	6.3	sequential	few GB/user	low	low	58GB per hundred users per day
Collaborative applications	4 – 8KB	N/A	sequential	N/A	low	low	10MB – 10GB per group (100 users) per day
Web email	4 – 8KB	2 – 3	sequential	100MB	low	low	10MB – 10GB per hundred users per day
Web groupware services	4 – 8KB	2 – 3	mixed	GBs	low	low	10MB – 10GB per hundred users per day
Database OLTP	$\sim$ 4KB	3	random	100GBs	high	low	1 – 10GB per TPCC test*
Database DSS	$\sim$ 32KB	8	sequential	TBs	low	high	100 – 800GB per test*
Multimedia	64 – 256KB	high	sequential	10TB	low	high	$\sim$ 10GB per hour
Backup and archival	large size	1	sequential	GB – TBs	low	low	GB – TB

Table 1: Workload characteristics of key applications. The Instructional workload refers to applications running on workstations on the UC Berkeley campus used for teaching undergraduate coursework [10]. Research refers to workstations used for research and development by computer science researchers in the same university [10]. Collaborative applications consist of group applications such as Lotus Notes. Web email refers to services such as Hotmail, Yahoo!, or Netscape WebMail. Web groupware services are web sites supporting virtual communities (discussion group bulletin boards, member profiles databases). OLTP stands for the online transaction processing workload. DSS stands for decision support systems. * The characteristics of these benchmarks depends largely on the size of the system. The numbers for TPC-C and TPC-H were reported in [9]. TPC-H results refer to RF1 from the PowerTest and a 116 warehouse benchmark run on on HP k-class server with 50 disks reported in [9]. Multimedia is a typical video-on-demand system serving tens of movies out of a total of a thousand movies.

3.1 Analysis of application traces

This section surveys several characterization studies that investigated the I/O behavior of filesystems and databases [3, 5, 10, 6, 9]. Not all studies used recent traces or traces collected during the same period. In fact, the trace collection period ranged from early nineties to late nineties. Nevertheless, such information is very useful in determining the high-level characteristics of various application classes. When exact information

is not available about the I/O behavior of a particular application class, a qualitative description is provided. This qualitative characterization is either based on published studies or direct investigation and experimentation by the authors. In either case, we specify the source of the characterization data when they are discussed.

We characterize an application workload by the following measures: (i) capacity requirements (total data set size), (ii) average request size in KB, (iii) access pattern (sequential or random), (iv) throughput requirements (requests/second), (v) bandwidth requirements (MB/sec), (vi) read vs. write ratio, and (vii) working set size.

3.1.1 Bandwidth requirements

The bandwidth requirements of an application dictate the bandwidth of the connection between customer server and storage location. From Table 1, it is clear that applications such as decision support systems consume tremendous amounts of bandwidth (in the hundreds to thousands of megabytes per second). Decision support systems perform a large number of full table scans and read large amounts of data. Such queries will be slowed down significantly if the bandwidth between the database server (customer site) and the storage system (storage location) is limited to typical communication bandwidths in the wide area (155Mbps $<$ 20MB/sec).

Other applications such as groupware, officeware, and research and instructional environments have low bandwidth requirements, and are more likely to perform acceptably when storage is remote, provided acceleration devices such as read and write caches are employed.

3.1.2 Throughput requirements

Most office, engineering, and collaborative groupware applications require limited I/O throughput. Database applications, both OLTP and DSS, on the other hand, require higher I/O throughput. OLTP makes a large number of small accesses (database records and log writes). DSS makes fewer but larger accesses to retrieve large portions of database tables to analyze them for potential patterns. OLTP therefore has a higher throughput (number of requests per second) requirement than DSS, while DSS has a higher bandwidth (total data transferred in megabytes per second) requirement.

3.1.3 Working set size

The working set size of the application is important because it determines the size of the cache required to subsume a large part of the application's requests. Database applications and streaming applications have large capacity and working set requirements. OLTP workloads have a footprint size that is dependent on the size of the back-end database. But most applications have a skewed access pattern (following a Zipf-like distribution) suggesting that the size of the cache can be relatively small compared to the back-end database.

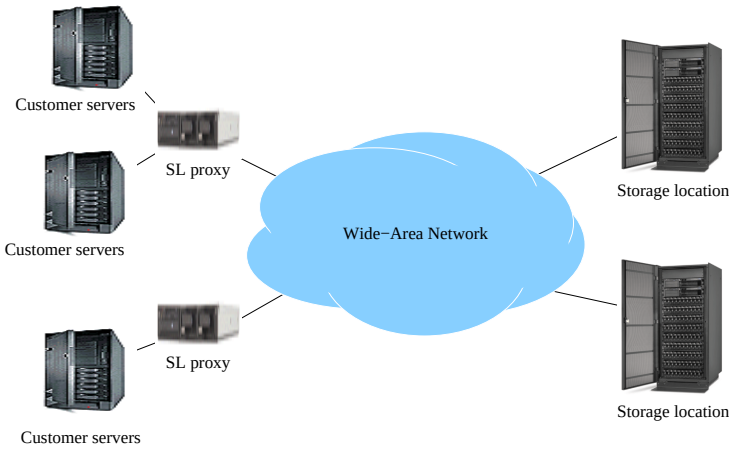

Figure 2: Storage proxy caching architecture.

3.2 Summary

Preliminary inspections of the workload characteristics summarized in Table 1 suggest that for data warehouses (data repositories used to answer DSS queries), it is more advantageous to host the entire data warehouse (DBMS, data mining software, and back-end storage) for a given customer at the service provider, rather than host only their block-level storage. This is because the database server executing the queries accesses storage intensively. In case of TPC-H (decision support benchmark), the bandwidth requirements are very high (hundreds of MB/sec). On the other hand, much less data is moved between the database client and the database server (e.g. Web accessible databases where the web client is the browser). Office/engineering, collaborative groupware and email applications, and possibly OLTP applications, have moderate bandwidth requirements. Furthermore, except for OLTP, the rest of these applications have moderate to low throughput requirements. They thus present good candidates for remote hosting.

4 Storage proxy caching

4.1 Overview of the storage caching architecture

To hide the latency of remote storage access, we propose to introduce intermediate caches as proxies close to the customer servers. These proxies are referred to as Storage Location proxies, or *SL proxies* for short. The SL proxy is a specialized appliance that performs caching, prefetching, and uplink bandwidth management. The goal of these proxies is to provide local-storage-access performance from remote storage. A customer server communicates with a remote storage location through the SL proxy. In particular, the SL proxy services the SCSI reads from the client by sending the object from its local cache when possible. Similarly, it services writes by writing them to local storage, replying to the client immediately. Data is then written back to the remote

storage in the background. The management of SL proxies would be provided either locally or remotely by the SL administrators. Figure 2 illustrates an example configuration. As shown in the figure, a single SL proxy can service multiple customer servers.

We assume the iSCSI protocol [7] is used for transmission of SCSI commands and data between the customer servers, SL proxies, and storage locations. The iSCSI protocol is a mapping of the SCSI remote procedure invocation model over the TCP protocol. TCP/IP is widely deployed and using it as a transport protocol allows for cost-effective commodity network components to be used. It is also a natural choice for remote storage services since most wide-area networks are predominantly IP networks. We do not model the details of iSCSI or TCP, and simply assume that iSCSI provides an ideal reliable communication channel between two nodes. Based on this abstraction, we investigate the amount of resources required to provide local-access levels of performance for the classes of workloads discussed above.

4.2 Performance of wide-area storage

This section uses a simple analytical model to estimate the efficacy of the SL proxy in improving the performance of wide-area storage services. To make the analysis tractable, we consider the simple case of a single customer server accessing a dedicated data volume hosted on a remote storage location. The SL proxy is deployed between the customer's site and the storage location. We assume the application has an I/O request stream described by a Poisson arrival process with parameter λ.

We denote by l_1 and b_1 the latency and bandwidth of the link between the customer server and the SL proxy. Likewise, we denote by l_2 and b_2 the latency and bandwidth between the SL proxy and the storage location. The service rates (i.e., the maximum throughput in terms of I/O requests per second) of the storage location and the SL proxy are denoted by U_s and U_p, respectively.

We compute the throughput-response time curve for three different storage configurations. The first is locally-attached storage with infinite SAN bandwidth and zero latency (*local*). The second is remote storage, accessed through a wide area network (*remote*). The third configuration is remote storage, accessed through the SL proxy located close to the customer's site (*cached*). Our analysis is based on simple mean value analysis, a technique used to solve closed queuing networks.

Throughout this section, we assume the parameters presented in Table 2 unless otherwise specified. We consider a storage system with 100 disks, each of which can perform 100 I/Os per second, or 10 ms per request. We also assume the workload to contain 20% writes and 80% reads, and the read hit rate at the SL proxy to be 70%.

Impact of distance

Figure 3 plots the effect of distance connected storage on application performance. The figure plots average I/O latency against throughput. The graph shows that for an OLTP like workload, the throughput of the wide-area link can constitute a performance bottleneck, causing the system to saturate before delivering the

Parameter	U_s	U_p	l_1	b_1	l_2	b_2
Value	10,000 IOs/s	10,000 IOs/s	0.1ms*	2Gbps †	5ms	155Mbps

Table 2: Wide-area storage parameters. * Zero delay in *local* configuration. † Infinite bandwidth in *local* configuration.

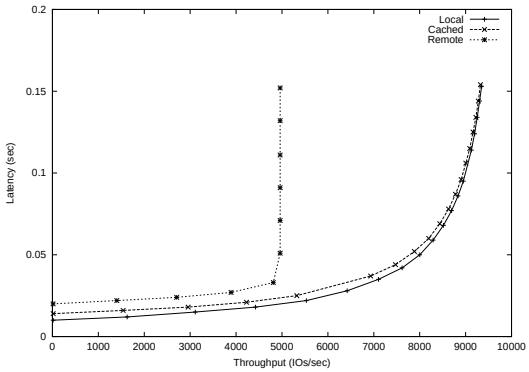

Figure 3: Impact of distance on client-perceived performance.

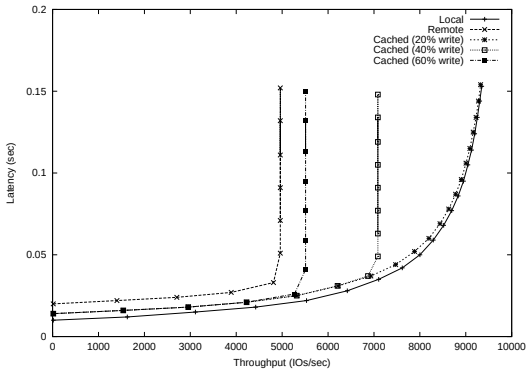

Figure 4: Impact of write traffic on client-perceived performance.

full throughput of the storage system. Caching at the SL proxy, however, can reduce uplink bandwidth usage, delivering the full throughput of the system.

Impact of read/write composition

Figure 4 plots the impact of write traffic on the performance of the storage cache architecture. A higher write traffic ratio implies higher uplink utilizations, since writes have to be eventually written back to the storage location. In this experiment, the write traffic ratio is varied from the baseline 20% to 60%. The graph shows that as the write traffic increases, the system saturates at a lower throughput.

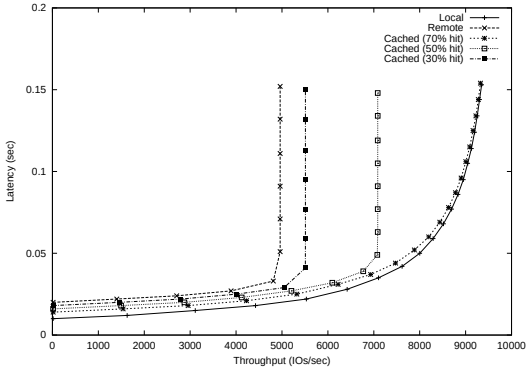

Figure 5: Impact of read hit ratio on cache performance.

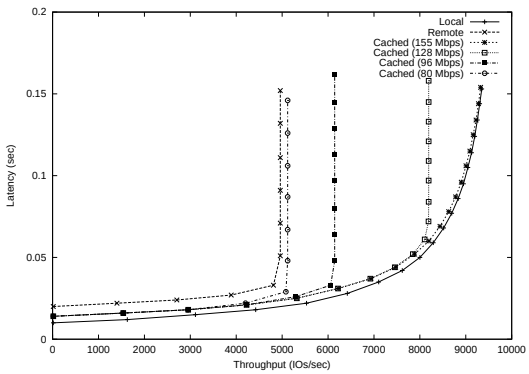

Figure 6: Uplink bandwidth reduction due to storage cache.

Impact of read hit rate

Figure 5 captures the impact of read hit ratios on cache performance. The graph shows that as the hit rate decreases from 70% to 30%, the system saturation point occurs at lower throughput. This is intuitive because with higher hit rate, more requests are serviced from the nearby proxy's cache, whereas lower hit rate results in higher uplink utilization, leading to the uplink (and the entire system) saturating at a lower I/O rate.

Bandwidth savings of storage cache

Figure 6 plots the throughput-latency graph for the three configurations with a relatively low bandwidth network connecting the cache to the back-end storage location. The caching appliance uses an uplink connection between the storage cache and the storage location with a bandwidth of 80 – 128 Mbps. In the *remote* configuration (without cache), the link bandwidth is 155 Mbps.

Note from Figure 6 that a *cached* configuration (with 70% cache hit rate and 20% writes) can achieve slightly better performance than a *remote* configuration while using close to half the uplink bandwidth.

4.3 Cache capacity planning

Our simple steady-state analysis delivers similar conclusions about the bandwidth requirements of uplink connections. In this section, we provide simple closed form steady state equations for the bandwidth and cache size requirements parametrized by the particular workload characteristics.

Uplink Bandwidth requirement

Assume that the SL proxy results in a hit rate h_r for reads. Similarly, denote the fraction of writes that are applied to a previously written (dirty) block that is already in the cache by h_{ovw}. Thus, only $1 - h_{ovw}$ of the writes are propagated by the SL proxy cache to the storage location. These values, of course, depend on the cache size and the write-back schedule. But as we will see, even optimistic but reasonable values for the hit rate suggest that substantial uplink bandwidth is required for remote high-throughput storage systems.

To bound the buffering requirement at the SL proxy, we want the upstream request bandwidth to be less than the back-end link capacity. Denoting by S the average size of the application I/O requests, the following equation must hold:

$$b_2 > (1 - h_r) * P_r * S * \lambda + (1 - h_{ovw}) * (1 - P_r) * S * \lambda, \quad (1)$$

where P_r is the read percentage, i.e., the ratio of read operations to the total requests for a given application. Equation 1 states that to achieve a throughput of λ, the uplink connection must contain sufficient bandwidth to carry the read misses and the write-back requests. Assuming reads and writes have the same hit ratio ($h_r = h_{ovw} = h$) the equation can be simplified to:

$$b_2 > (1 - h) * S * \lambda \quad (2)$$

Equation 2 states that the bandwidth capacity of the uplink, b_2, is proportional to the bandwidth requirement of the application, with the proportionality factor equal to the miss rate $(1-h)$. This implies that even if the hit rate is quite high, say 80%, then b_2 must be at least 20% of the bandwidth requirement of the application when using locally attached storage. From Table 1, it is clear that DSS and OLTP require substantial bandwidth, especially DSS, which would make the cost of the uplink quite prohibitive.

Cache throughput requirement

From Table 1, it can be seen that email, collaborative and office/engineering workloads have low bandwidth requirements which can be satisfied using an uplink bandwidth with the capacity of a single or a few OC-3 connections. For these applications, however, long-term bandwidth is not the only performance requirement. Latency of storage access using the proxy cache must be comparable to that of local storage. Latency to the client is affected largely by the design and size of the SL proxy. Another equation must hold to hide degradation from applications. The SL proxy should be capable of handling the short-term request bursts of the clients. To achieve a request throughput of λ, the caching appliance must contain enough throughput to handle the client writes, i.e., the following equation must hold:

$$U_{c,write} > (1-P_r) * \lambda \tag{3}$$

Equation 3 states that the write requests originating from the clients have an arrival rate of $(1-Pr) * \lambda$, since $(1-P_r)$ is the fraction of the total client requests that are writes. Thus, the SL proxy must have a write throughput that is commensurate with that of locally attached storage. Otherwise, some writes will notice remote latencies (l_2). This requires the local write throughput to be a good fraction of the write throughput of the customer's would-be local storage array. Such a throughput in a small appliance is possible only with NVRAM. Although the price per megabyte for NVRAM is one to two orders of magnitude greater than commodity magnetic disks, we find that a small amount of NVRAM can suffice to provide performance levels commensurate with locally attached storage.

5 Conclusions

The feasibility of hosting back-end storage for data-intensive applications such as decision support (data mining) relies heavily on the availability of cost-effective high-bandwidth (multi-Gigabit) links between the customer site and the storage location. In particular:

- Remote storage access is feasible for collaborative group applications, office/engineering type workloads, multimedia streaming, but not for database applications, especially decision support systems.
- OLTP applications may tolerate remote storage access if provided with cache accelerators for reads and writes and if the working set is relatively small. But the database server accesses storage intensively

in case of a DSS query, for example, (performing many full table scans) to compute the query result. Thus a more attractive hosting model for databases may be entire application hosting (DB+storage). This is because the query supplied by the user to the database and the returned results are usually small. Therefore it is better to split the application at the DBMS client / DBMS server level, rather than at the DBMS server / storage level.

References

[1] Bandwith Market, Ltd. Bandwidth Market Releases Extremely Low Internet Access and Bandwidth Rates, August 2001. http://www.bandwidthmarket.com/news/pressrelease01_08_30.html.

[2] T. Clark. *Designing Storage Area Networks*. Addison-Wesley, 1999.

[3] P. Couvee. I/O Performance Considerations on Bull Escala AIX Servers, January 1998. http://www-frec.bull.com/docs/wp_ioperf.htm.

[4] Epoch Partners. Metro Optical Networking Providers: Bringing Light to the (Metro) Masses. http://www.epoch.com.

[5] W. Hsu, A. J. Smith, and H. C. Young. I/O Reference Behavior of Production Database Workloads and the TPC Benchmarks - An Analysis at the Logical Level. Technical Report RJ 10166, IBM Almaden Research Center, San Jose, CA, November 1999.

[6] W. Hsu, A. J. Smith, and H. C. Young. Analysis of the characteristics of production database workloads and comparison with the TPC benchmarks. In *IBM Systems Journal*, volume 40, 2001.

[7] Internet Egineering Task Force. ISCSI draft (Version 9), November 2001. http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-09.txt.

[8] Network Appliance. F800 Enterprise Filers and Data ONTAP Operating System. http://www.netapp.com/products/-filer/f800_ds.html.

[9] E. Riedel, S. Spence, and A. Veitch. When Local Becomes Global: An Application Study of Data Consistency in a Networked World. In *Proceedings of the 20th IEEE International Peformance, Computing and Communications Conference*, april 2001.

[10] D. Roselli, J. Lorch, and T. Anderson. A comparison of file system workloads. In *Proceedings of the 2000 USENIX Annual Technical Conference*, San Diego, CA, june 2001.