

ACM Computing Surveys **28**(4es), December 1996, <http://www.acm.org/pubs/citations/journals/surveys/1996-28-4es/a210-muntz/>.
Copyright © 1996 by the Association for Computing Machinery, Inc. See the [permissions statement](#) below. This article derives from a position statement prepared for the [Workshop on Strategic Directions in Computing Research](#).

System-level Design Issues for Storage I/O

[Richard R. Muntz](#)

Computer Science Department, U.C.L.A.
3277A Boelter Hall, Los Angeles, CA 90095-1596, USA
muntz@cs.ucla.edu, <http://www.cs.ucla.edu/~muntz/>

[Joseph Pasquale](#)

Dept. of Computer Science and Engineering, U. C. San Diego
La Jolla, CA 92093-0114, USA
pasquale@cs.ucsd.edu, <http://www-cse.ucsd.edu/~pasquale/>

Abstract: I/O systems are becoming more complex and must be designed by considering the entire system, end to end. We make a number of recommendations to address this problem, including the following. (1) More emphasis is needed on tertiary storage and on the whole (multilevel) storage hierarchy in general. (2) We must pay more attention to issues of resource management and availability in the network, especially if network-attached storage devices become more viable. (3) To improve performance, the operating system must give user-level processes more control over the data path between the storage device and the process, or be able to accept and exploit high-level hints about the application's behavior and its most important performance metrics/quality of service. (4) Finally, more emphasis should be placed on content-based or semantic-based compression, where we believe the greatest advances remain ahead of us.

Categories and Subject Descriptors: D.4.2 [**Operating Systems**]: Storage Management - *storage hierarchies*; D.4.4 [**Operating Systems**]: Communications Management - *input/output, network communication*; B.4.2 [**Input/Output and Data Communications**]: Input/Output Devices - *disks, channels and controllers*; E.4 [**Data**]: Coding and Information Theory - *data compaction and compression*;

General Terms: Algorithms, Design, Management, Measurement, Performance.

Additional Key Words and Phrases: I/O, communication.

1 I/O Architectures

I/O can no longer be viewed from the perspective of what happens between an I/O device and the machine it is connected to. I/O systems are becoming more complex, and must be designed by considering the entire system, end-to-end. For example, storage systems are themselves distributed systems, comprised of a hierarchy of storage devices of different speeds and sizes connected by (different types of) networks.

Recommendations:

- More emphasis is needed on tertiary storage and on the whole (multilevel) storage hierarchy in general. It seems that most work has concentrated on the management of two "adjacent" levels (usually main memory and disk).
- We need to pay more attention to work in communication networks. There are many similarities in the type of resource management that is done in storage and networks. Furthermore, storage systems should be more aware of what's going on in the network (e.g. in terms of resource availability), especially if network-attached storage devices become more viable.

2 OS-Related Issues

Operating systems are getting more and more "in the way" between the user and the storage system. The buffering and caching done by the OS may actually be detrimental to performance.

Ultimately, the application (or, more typically, a server or middleware) should be provided with more control over low-level functions and let do what it thinks best. A good example is the old story about letting database systems control their own buffering because they know best how to do it, rather than having the OS try to do it.

Recommendations:

- One approach is that the OS give user-level processes more control over the data path between the device and the process (e.g. whether the OS does buffering, and if so, how much; whether one can control if data ends up cached, where the caching occurs, how is it controlled). Going a step further, the OS may provide the ability to have data flow directly from one device to another if a device-to-device transfer is what the process ultimately wants to accomplish (e.g. a file server that gets data from disk and puts it out to the network device).
- Another approach is to consider how operating systems can accept and exploit high-level hints (e.g. pragmas) about the application's behavior and its most important performance metrics/quality of service. Getting this right will be difficult, e.g. determining what the hints should be or how the application should specify quality of service. But if done right, the result is a powerful system in which the application provides information only it best knows about, and the OS uses this information to manage the underlying resources, doing what it knows best about, i.e. managing the state of the resources and the global demands upon them.
- Finally, it is important to identify particular application categories, e.g., (1) real-time continuous media, (2) scientific computing, and (3) databases, and consider how the above-mentioned I/O issues

apply. Indeed, some issues are substantially different across the different application domains. For example, QoS issues in real-time continuous media are quite different from those in the other applications.

One especially important and interesting research challenge is how best to support a mixture of application workloads. One version of the question is to what extent one can build one storage system that can be statically configured for each workload type. (In other words, the goal of this approach is limited to achieving software reuse.) Another, harder version of the question is how to build a storage subsystem that can concurrently support a mixed workload of applications, addressing their individual requirements (throughput, latency, jitter, etc.) without a priori partitioning of resources. Most work to date is quite restrictive and assumes that only one class of workload is present. There are a few exceptions, such as some designs of VOD systems in which some consideration has been given to including non-real-time workload, but these are scarce.

3 Data Compression

How will future data-compression techniques influence I/O and vice versa (in addition to simply reducing bandwidth and storage requirements)? For example, lossy compression is influencing I/O in requiring variable retrieval rates. What about the future? What will be the effects of important schemes currently being researched, such as content- (or object-) based compression rather than the more common pixel-based compression (like JPEG). How (if at all) will this influence storage and retrieval?

Recommendations:

- Content-based or semantic-based compression is a very exciting area, perhaps the one where the greatest advances remain ahead of us. This area should definitely be emphasized as strategically important.

4 Understanding Device Technology Trends

Finally, researchers need a better understanding of where device technology is likely to go and how it will affect I/O problems five and 10 years from now. Will holographic memories make it? What exactly will they look like? Will semiconductor memories overtake disk in price per MB? Will that make solid-state disks the dominant secondary-storage device? For DVD technology, what will be the price tradeoffs? Will the technology stay at a plateau for 20 years after reaching the "blue-light special" capacity of 40GB and a MB/sec bandwidth?

Recommendations:

- This information is difficult to come by. This is not a "research direction" but could help target research toward important topics and away from soon-to-be obsolete issues. We recommend wider dissemination of such information through explicit workshops or components of existing conferences.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To

copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., fax +1 (212) 869-0481, or permissions@acm.org.

Last modified: Mon Feb 24 15:11:37 EST 1997

Joseph Pasquale <pasquale@cse.ucsd.edu>