

Impact

June 30, 2000

Solid Data Systems: File Caching to Multiply Storage and System Performance

Performance-robbing I/O waits may bring essential transaction-processing systems to their knees. With a Solid Data Excellerator file caching system to cache frequently accessed files, a Net-economy-based enterprise can substantially increase storage-related application performance at lower cost.

Net Economy Demands High Storage Performance

Net economy enterprises — both rising dot.coms and veteran Fortune 2000 enterprises — demand high performance from their new transaction-intensive applications: e-mail (e.g., Internet Sendmail); messaging (e.g., IBM's MQSeries); online trading; and Customer Relationship Management (CRM) systems among them. Unfortunately, too often the servers that drive the transaction process are unable to perform to their potential because of excessive I/O waits. Under-performance — i.e., longer than reasonable user response time — is unacceptable. Dissatisfied or lost users will result, as will lost revenue.

File Caching May Be the Answer

Using a Solid Data Systems Excellerator, a solid-state disk system, for file caching may solve the storage-related performance problem. File caching is effective where the I/O waits are due to file skewing. File skewing means that a small percentage — say 5% — of all files are responsible for the majority of I/O waits. File caching of the “hot” files may

significantly increase performance. Of course, file caching using DRAMs costs money (no free lunch), but the cost increases may run less than the cost of the alternatives.

Add Servers, Disks, or File Cache?

Typically, an IS organization would consider two alternatives to the I/O wait problem: adding extra servers to get more CPU cycles, or adding more disks to increase the number of I/Os per second (IOPS) available for servers to process. The first alternative brings added cost as well as the burden of configuring and managing them.

The second alternative may require many more disks, because each new disk adds relatively few new IOPS. Moreover, the extra disks may require a new disk array — at significant added cost — and necessitate expensive, skilled IT staff to tune and re-tune the data placement.

If the I/O wait is not a result of a skewed file distribution, one (or a combination) of those two alternatives to file cache may be the answer. Otherwise, an Excellerator file caching system not only

solves the I/O-wait problem, but also has non-disruptive scaling capability. Non-disruptive scaling means no downtime; no labor-intensive, productivity-bleeding reconfiguring; and no risk of expensive replacement of existing investments.

Independent Scaling in a SAN

Scaling is especially important when a Net economy application requires a mix of both clustered servers and a storage area network (SAN) to deliver fast response time (high performance) and 24x7 uptime (high availability). File caching provides independently scalable performance in a SAN so file-caching Excellerators can be added without impacting the other components of the SAN. That is because an Excellerator FC (Fibre Channel) solid-state disk acts as a file cache appliance. The appliance provides a specialized service — caching — as a stand-alone device. Changes such as upgrades, modifications, and removals to servers or to RAID systems do not affect the file cache appliance.

File Cache Versus Block Cache

Through an analysis of an application's file structure, the files to be placed in cache (such as message queues and database log files) can be targeted precisely. Solid Data provides software to aid that intelligent selection process — e.g., I/O Dynamics for Oracle hot file identification. In contrast, block caching — typically RAID controller cache — applies algorithms that estimate what blocks of data to keep in cache. For a skewed file distribution, a file cache is

very likely to be right; a block cache may only have a good chance of being right.

Where Should File Cache Be Put?

A solid-state disk is not the only way to file cache. A portion of server memory or disk controller cache can be reserved for that purpose. In contrast to both options, Excellerator has the advantage of independent scaling. An Excellerator system has built-in backup protection to flush cache to non-volatile disk in case of a power failure; typically, server memory does not have the same level of protection. In the case of disk controller cache, there may not be enough cache available to meet the needs of both block and file cache.

A Family Affair

Solid Data offers a family of Excellerator file caching systems that range from 536 MB to 17 GB in capacity, and that support Fibre Channel and UltraSCSI connections. The Excellerator family works with Windows NT, HP-UX, and Sun Solaris servers. The total cost depends on cache size and varies over a wide range — from just under \$10K to more than \$100K.

Aberdeen Conclusions

An Excellerator file caching system can cost-effectively alleviate sub-par application performance resulting from I/O waits. A Net economy enterprise should consider an Excellerator file caching system so that its revenue-producing application does not have to (I/O) wait anymore.

— David Hill