Article
Murray Oldfield · Jun 17, 2016 2m read

# Data Platforms and Performance - Part 6 Caché Storage IO Profile

Myself and the other Technology Architects often have to explain to customers and vendors Caché IO requirements and the way that Caché applications will use storage systems. The following tables are useful when explaining typical Caché IO profile and requirements for a transactional database application with customers and vendors.  The original tables were created by Mark Bolinsky.

In future posts I will be discussing more about storage IO so am also posting these tables now as a reference for those articles.

---

A list of other posts in this series is here

---

It is vital to have well set up storage such as a storage array to provide predictable disk IO performance, support high availability features and provide storage redundancy, scalability, and reliability for your applications.

## Caché Storage IO Profile

Depending on the storage array technology selected, disk storage at the physical layer may be separated into pools or disk groups. If possible for availability and performance storage should be partitioned based on the expected IO profile. For example data and journals (transaction logs) should be on separate physical disk groups because they have different IO profiles and also for availability as corruption in the data disk group can be recovered using the journals in a separate disk group isolated from the corruption.

Similarly backups should be in a separate disk group. There are many choices of storage configuration depending on operating system, storage vendor and array model. Exact requirements will be application specific with special attention required for physical and logical configuration including allocation of physical and logical groups or pools, RAID types, filesystem types and concurrency, GB space allocated, and so on.

The read and write IO profile of Caché is detailed in the following table:

| IO Type | When | How | Notes |
|---|---|---|---|
| Database Reads | Continuous by user processes | User process initiates a disk I/O to read the data | Database reads are performed by either the daemons serving web pages, SQL queries, or direct user processes. |
| Database Writes | Burst approx. every 80 seconds or percentage of database cache pending updates | Database write daemons (up to 8 processes) | Database writes are performed by a set of database system processes known as write daemons. User processes update database cache and a trigger (time or activity threshold) will send the updates to disk using the write daemons. Typically expect anywhere from a few MBs to several GBs that need to be written during the write cycle depending on transaction rates. |
| Journal Writes | <2secs, full journal buffers, or sync request | Database journal daemon (1 process) | Journal writes are sequential and variable in size from 4KB to 4MB. There can be as low as a few dozen writes per second to several thousand per second for very large deployments using ECP and separate application servers. |
| Write Image Journal Writes | Burst approx. every 80 seconds or percentage of database cache pending updates | Database master write daemon (1 process) | This journal is used to protect physical database file integrity from system failure during a database write cycle. Writes are approximately 256KB each in size. |

## Caché Storage IO Requirements

I find that bottlenecks in storage are one of the most common problems affecting database system performance. A common problem is sizing storage simply for GB capacity, rather than allocating a high enough number of discrete disks to support expected Input/Output Operations Per Second (IOPS). Although SSDs and tiered storage are now more common care must be taken to ensure IOPS are available.

To guarantee acceptable response times for end users a disk array with a minimum IO performance profile is required. The requirements vary slightly depending on whether separate Application (ECP) servers are used. The following table details the expected storage response times and notes on IO profile.

| IO Type | Average Response Time | Maximum Response Time | Notes |
|---|---|---|---|
| 8KB Database Random Read (non-cached) | <=6 ms | <=10 ms | Database blocks are a fixed 8K – most reads to disk will not be cached because of large database cache on the host. |
| 8KB Database Random Write (cached) | <=1 ms | <2 ms | All database file writes are expected to be cached by the storage controller cache memory. |
| 4KB to 4MB Journal Write (without ECP) | <=1 ms | <=2 ms | Journal writes are sequential and variable in size from 4KB to 4MB. Write volume is relatively low when no application servers are used. |
| 4KB to 4MB Journal Write (with ECP) | <=0.5 ms | <=1 ms | Journal synchronisation requests generated from ECP impose a stringent response time requirement to maintain scalability. The synchronisation requests issue can trigger writes to last block in the journal to ensure data durability. |

## Whats missing?

Before you talk to your vendor you will need to have estimates of the IOPS you expect your application to be driving so that the storage options can be configured to meet the requirements above. Part 1 and 2 of the series show examples of how to collect metrics.

#Best Practices #InterSystems Business Solutions and Architectures #InterSystems Data Platform Blog #Performance #System Administration #Caché #InterSystems IRIS #InterSystems IRIS for Health

**Source URL:** https://community.intersystems.com/post/data-platforms-and-performance-part-6-cach%C3%A9-storage-io-profile