

# FAST700 and AIX 5.2: Performances Tests using iozone

## Introduction

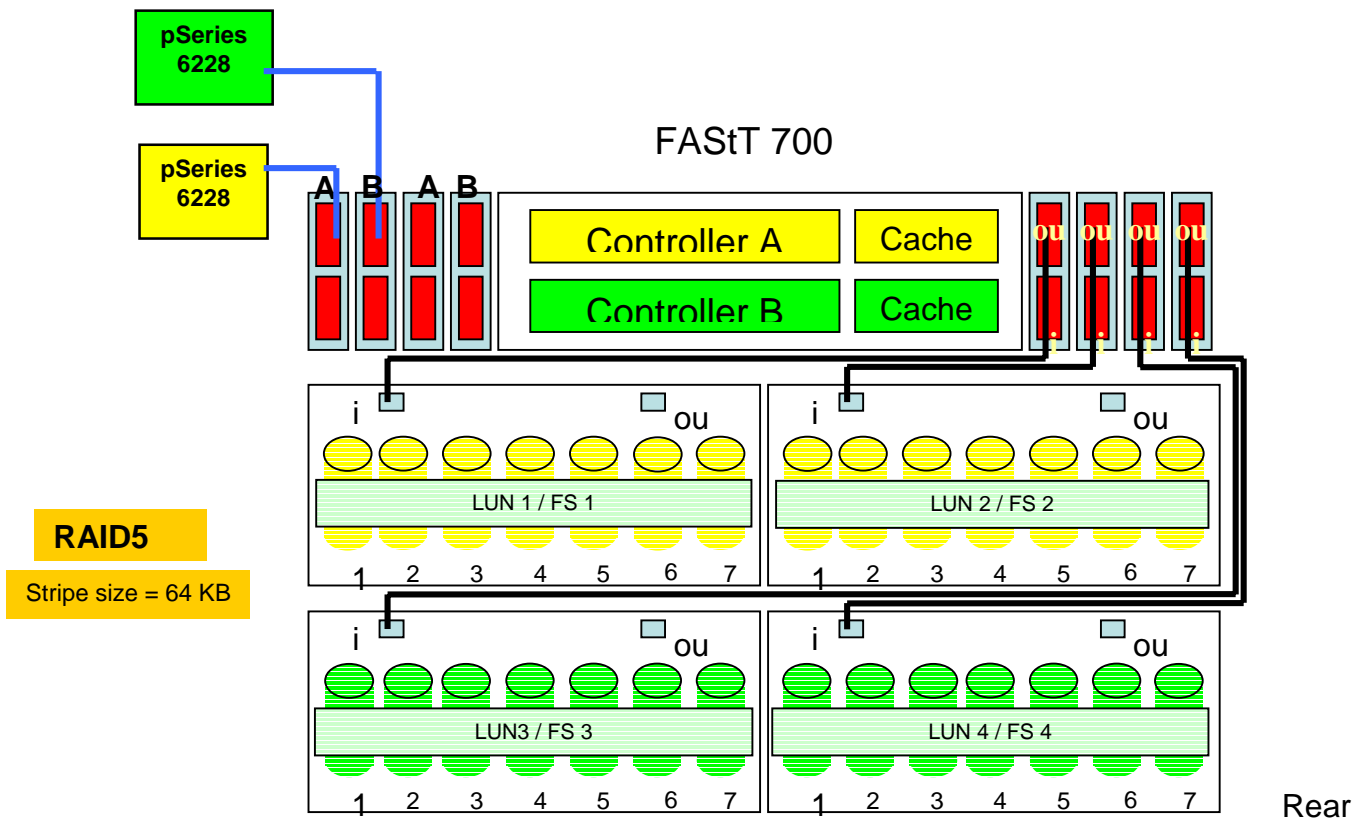
We've just finished a test with the FAST700 storage server and AIX for the customer CINES. The customer asked for a benchmark to measure sequential write performance on a FAST700 in a pSeries environment using iozone tools (v172).

The purpose of this document is to give a feedback on the performance results and some information on the FAST / pSeries implementation.

## 1) Performance Results

### Hardware environment

- 1) One FAST700 with 2 EXP700
- 2) One H80 6 CP, 14 GB memory, 2 FC card
- 1) 2 raid arrays RAID5 created per EXP (7disks per RAID array)
- 2) 1 LUN per array so a total of 4 LUNs



## The tests

We ran the tests with 2 different configurations:

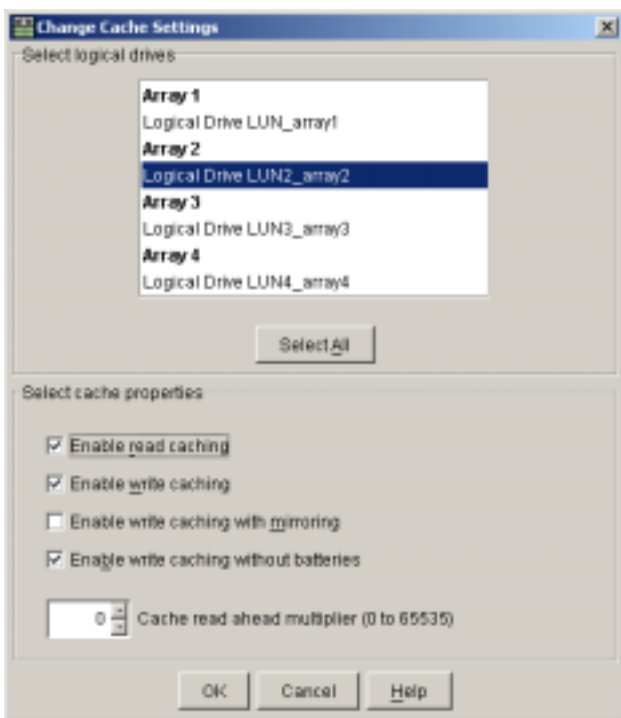
- 1) One test using one FAStT700 and 2 EXP700-36 GB disks (One EXP at 15krpm, the other one at 10krpm) connected to a pSeries H80 through 2 FC adapter (6228)
- 2) The second one using 2 FAStT700 and 2 EXP700/FAStT connected to the H80 through 4 FC adapter (6228)

We focused on the first configuration (with only one FAStT) in order to verify the maximum throughput that can be obtained for sequential write operations with one FAStT storage server and 2 EXP.

With the second configuration (2 FAStT700), we reached the H80 CPU limits before reaching the 2 FAStT storage servers limits and therefore didn't managed to get more than 334 MB/s with 2 FAStT700 storage servers.

We tried several FAStT cache parameters, and get the better performances with the following ones:

- a. FastTwrite cache without mirroring

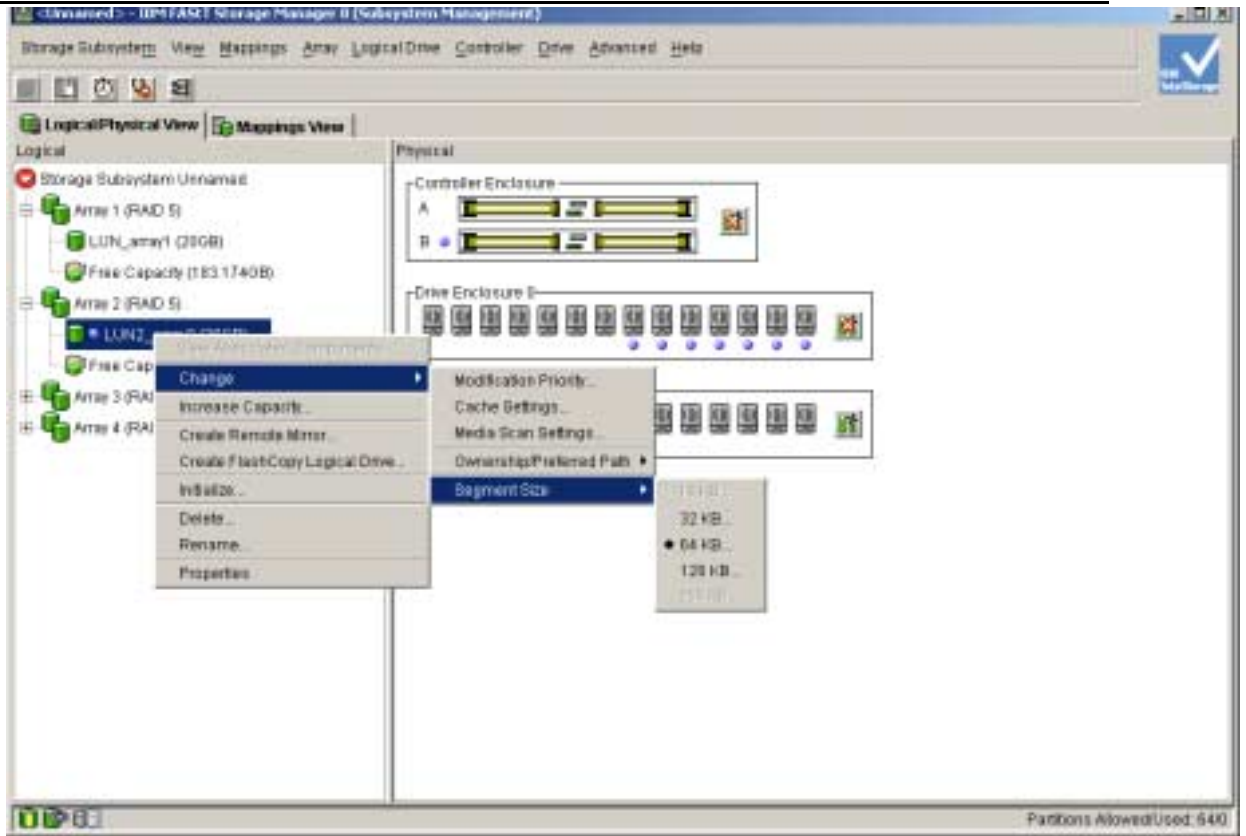


It is the "Write Back Mode".

When controller handling a write request:

1. The controller writes the data to the Cache
2. The controller returns a command complete notice, even though data has not yet been sent to the drive.
3. The controller sends the data to the drive.

- b. Segment size = 64KB



The segment size or stripe size is the unit of storage used by the RAID controller to interleave data between different drives.

c. Cache segment = 16KB

This is the size of cache memory allocation unit and it can be either 4K or 16K.

**AIX environment:**

We used 4 filesystems (JFS type not stripped) and 4 files for the test, each filesystem on a different array. We mounted the filesystem using the “release behind write” options using the following command:

```
> mount -o rbw
```

The customer asked for one test with iозone tools to generate intensive sequential write operations with a 64KB block size.

**IOzone** is a filesystem benchmark tool. The benchmark generates and measures a variety of file operations (*Read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read, pread, mmap, aio\_read, aio\_write*)

We used the following iозone command for the tests:

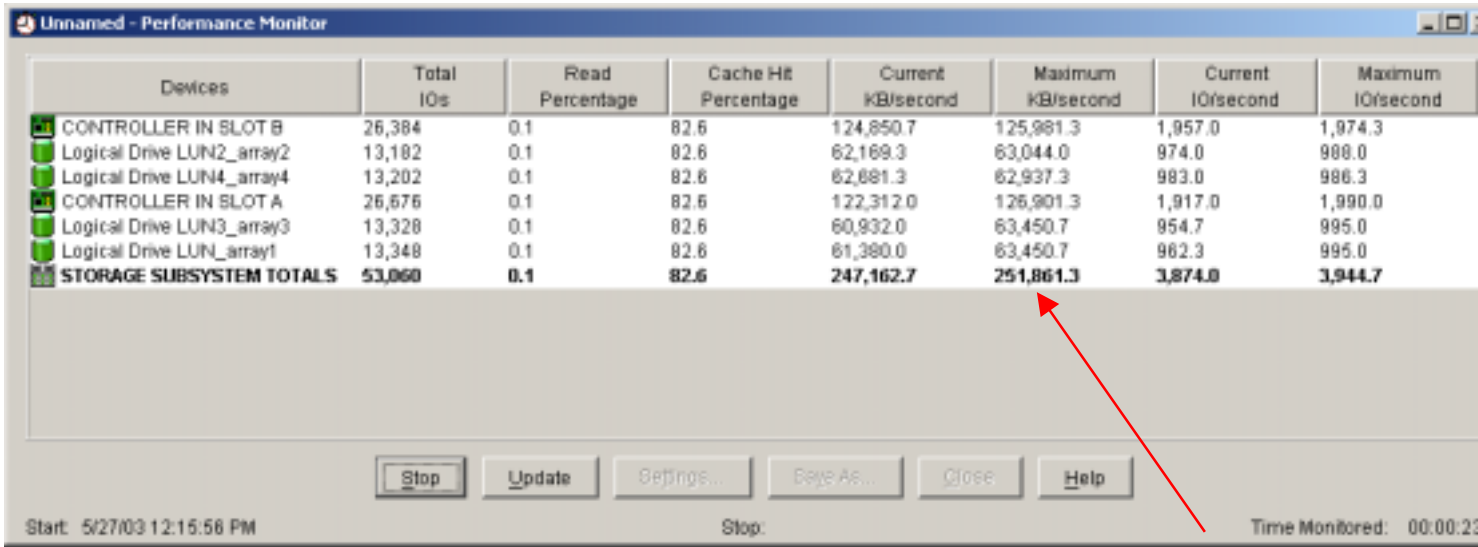
```
➤ iозone -r64k -R -s10g -i0 -t4 -F /dir1/fic1 /dir2/fic2 /dir3/fic3 /dir4/fic4
```

This command starts 4 parallel write processes. Each process write a 10 GB files.

Here is the summary of the results:

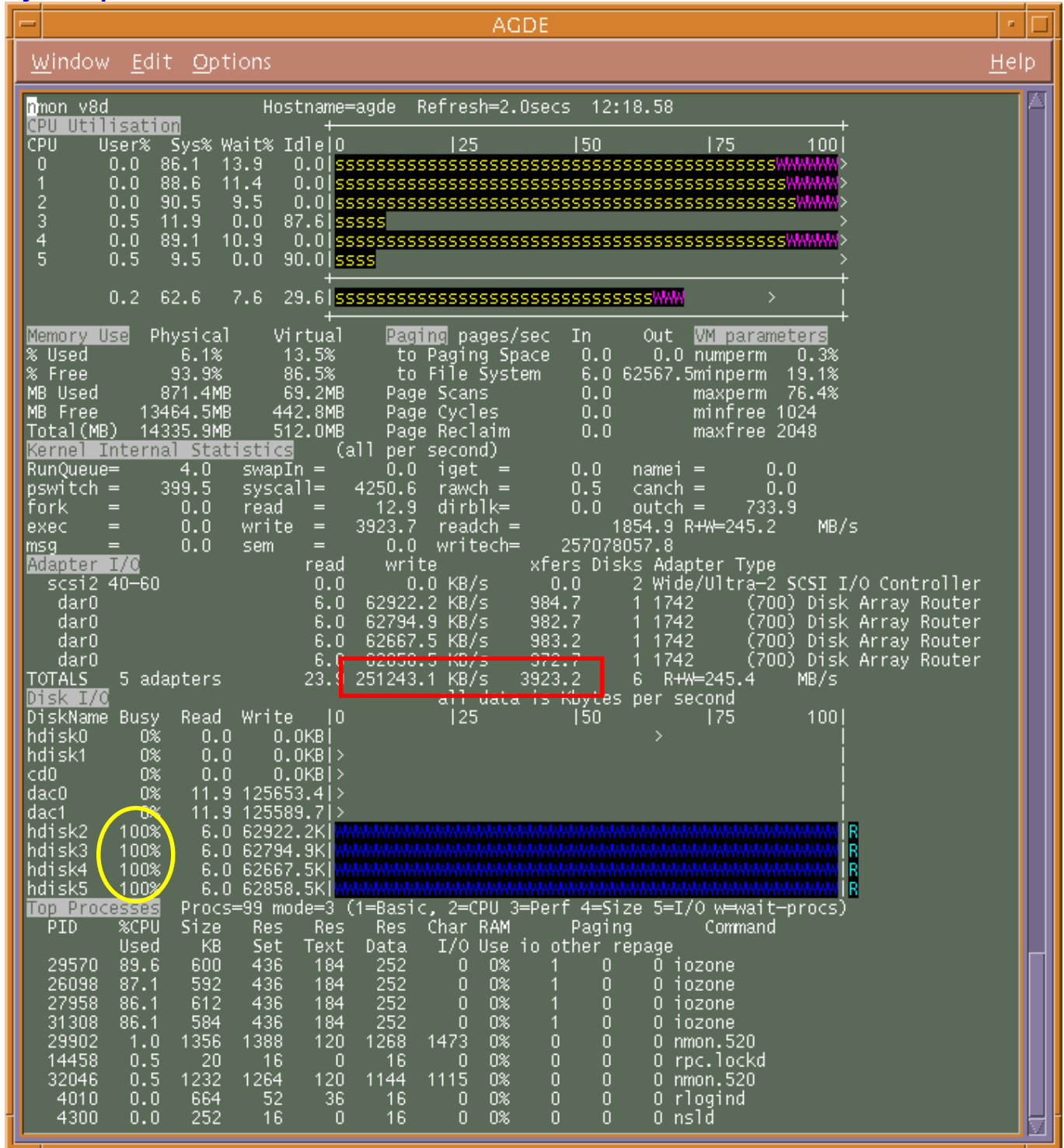
Tests	Cache mode	Num. of EXP	sys	idle	wait	Performances
<b>1 FAStT</b> Iozone -r64k -R -s10g -i0 -t4 -F file1 file2 file3 file4	Write back 80-80	2 per FAStT	20 (write)	30	50	<b>258 MB/s</b>
<b>2 FAStT</b> Iozone -r64k -R -s10g -i0 -t8 -F file1 file2 file3 file4 file5 file6 file7 file8	Write-back 80-80	2	99 System overloaded	0	0	<b>334 MB/s</b>

FAStT performance monitor results :



Max Write perf

**System performance:**



## Conclusion :

The maximum we obtained from one FAStT700 + 2 EXP700 was 258 MB/s for sequential write using a 64KB block size.

After adding another FAStT700 + 2 EXP700 we reached a maximum of 334 MB/s.

We reached the limit of the pSeries CPUs before reaching the limit of the 2 FAStT700 storage servers.

## 2) Additional informations

Before starting any (performances) tests, you should verify the following components:

### 1) On the pSeries:

- a. Verify the “devices.fcp.disk.array.rte” fileset level (FASTT driver)
  - i. 5.2.0.10 in our case.
- b. Verify devices.pci.df1000f7.rte level (6228 driver)
- c. Verify devices.pci.df1000f9.rte level (6228 driver)
- d. Check the FC card type and the FC card microcode level using “lscfg -vl fcsx” command

```
{agde:root}/etc/microcode -> lscfg -vl fcs0
fcs0          U0.1-P1-I1/Q1 FC Adapter
```

```
Part Number.....00P4494
EC Level.....A
Serial Number.....1D3080C154
Manufacturer.....001D
FRU Number..... 00P4495
Network Address.....10000000C931A2E1
ROS Level and ID.....02C03891
Device Specific.(Z0).....2002606D
Device Specific.(Z1).....00000000
Device Specific.(Z2).....00000000
Device Specific.(Z3).....02000909
Device Specific.(Z4).....FF401050
Device Specific.(Z5).....02C03891
Device Specific.(Z6).....06433891
Device Specific.(Z7).....07433891
Device Specific.(Z8).....20000000C931A2E1
Device Specific.(Z9).....CS3.82A1
Device Specific.(ZA).....C1D3.82A1
Device Specific.(ZB).....C2D3.82A1
Device Specific.(YL).....U0.1-P1-I1/Q1
```

- e. Check the FC card parameters
- f. Check the FC card positioning (slots)

During the benchmark, we observed different throughput on the FC adapters due to the wrong card placement on the H80 PCI bus:

fcs0 1-P1-I3 66MHz (64 bits)  
fcs1 1-P1-I6 33 MHz (32 bits)  
fcs2 P1-I9 66 MHz  
fcs3 P1-I11 33 MHz

In order to avoid this kind of problem, verify the FC card position on the PCI bus using the following:  
>lsslot -c pci

Place the 6228 FC card in 66Mhz slot respecting the rules detailed in the “PCI Adapter Placement Reference” guide in order to avoid performance degradations.

## 2) Here are some verifications to be done on the FAStT storage server :

- a. Verify the cabling between the FAStT controller and the EXPs, and the FAStT controller and the pSeries
- b. Control the Gigabit switch position (2Gb led should be green) on the FAStT controllers (host side and disk side) and on the EXPs
- c. Clean the previous configuration, use <SysWipe> command on the serial interface, wait for completion message and then type <SysReboot> on each controller. (The password to connect to the FAStT serial interface is infiniti)
- d. Configure the Ethernet port used to connect the FAStT controllers to the Ethernet network to: “Auto negotiate”
- e. Verify the FAStT microcode level on the two controllers and on the EXPs

Depending on which microcode level is installed on the FAStT storage server, it may be necessary to change this microcode level, by using either the Storage Manager Interface or the serial link. The current version of the fibre channel disk subsystem firmware can be downloaded from the IBM Support site: <http://www.pc.ibm.com/us/support/>

- f. create LUNs and hosts
- g. delete LUN31 for AIX hosts

After configuring the LUNs and host attachment.

## 3) Check you can access the disk from the pSeries

- a. Cfgmgr on the pSeries
- b. >fget\_config -A to list the number of LUN per controller

The AIX devices are:

fcs0 : fiber channel adapter

fscsi0 :logical device for SCSI protocol over FC

dar0 : disk array router -logical device -one per FAStT

dac0,dac1: disk array controllers (two per FAStT)

hdiskx: logical disk

**Be careful !**

\* The cache parameters (write/read disable/enable, mirroring, read prefetch ...) are reset to default at each server reboot or “mkdev” command.

For our tests, we saved the parameters on the FAStT storage manager interface and checked and restored the parameters before the different runs.

\* If you have more than 1 EXP in your configuration, verify you have different address on the EXPs to avoid any address conflict.

\* In case of defective batteries, the FAStT write cache will not be operational except if you have selected option “enable write caching without batteries” in the FAStT cache parameters.