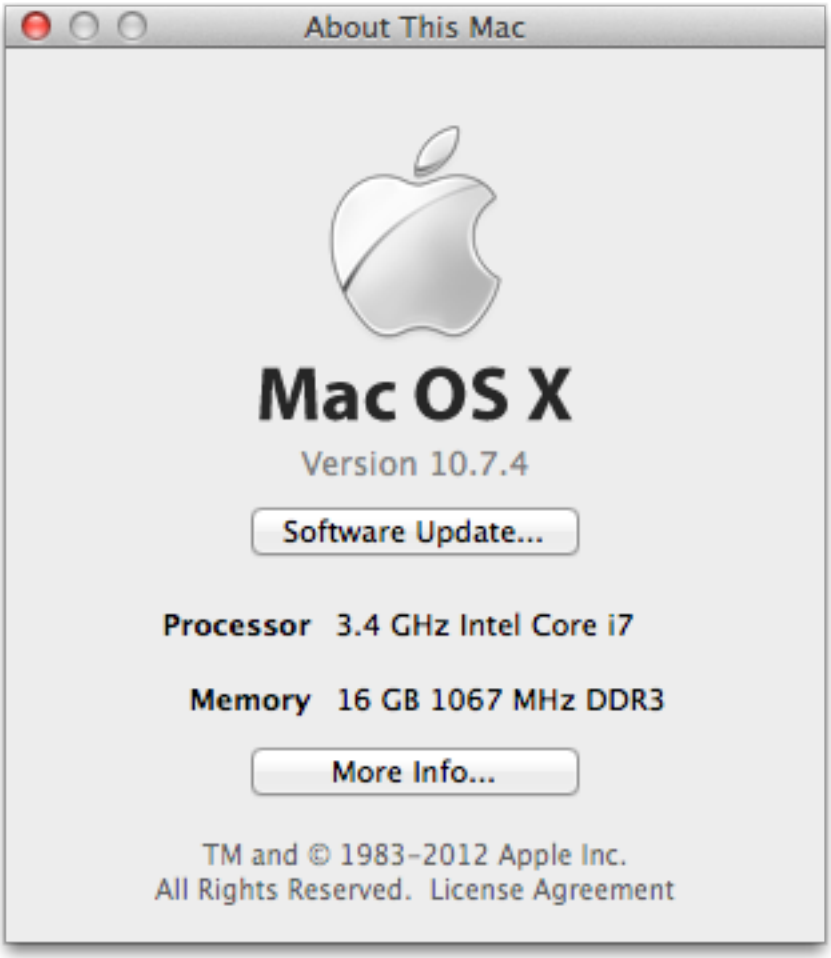
Introduction

This document summarises the results of some performance testing I carried out on a Pegasus R6 Thunderbolt RAID drive from Promise Technologies Inc.

Test Environment

The unit I used was a Pegasus R6 Thunderbolt RAID from Promise Technologies Inc. It is fitted with 6 drives - each 1TB capacity (Hitachi HDS72101)

I hosted the RAID array on an iMac with a 3.4GHz Intel Core i7 CPU with 16GB of RAM and running OS X 10.7.4



Version 10.7.4

Processor 3.4 GHz Intel Core i7

Memory 16 GB 1067 MHz DDR3

TM and © 1983-2012 Apple Inc.
All Rights Reserved. License Agreement

Method

I originally intended to use the Blackmagic Disk Speed Test software (available for free from Apple's App Store) to run the tests, but the results were very inconsistent at the levels of performance this RAID array achieves, so I devised my own method using simple, command line tools to test sustained throughput. Note I am not particularly interested in random I/O and did not test this.

Basically I configured the RAID using the utility supplied, into many different combinations and then mounted the resulting array and tested it using the following method:

1. Mount disk as HFS+ journalled filesystem using DiskUtility

2. Use "`dd`" command line tool to write 100,000 blocks of 1MB each onto the Pegasus (100GB) in the following manner:

   ```
   dd if=/dev/zero of=PegasusFile.junk bs=1024k count=100000
   ```

3. Use OS X "`purge`" command to flush the disk cache

4. Use "`dd`" to read the file back from the Pegasus and discard it to /dev/null but time the operation like this:

   ```
   dd if=PegasusFile.junk of=/dev/null bs=1024k
   ```

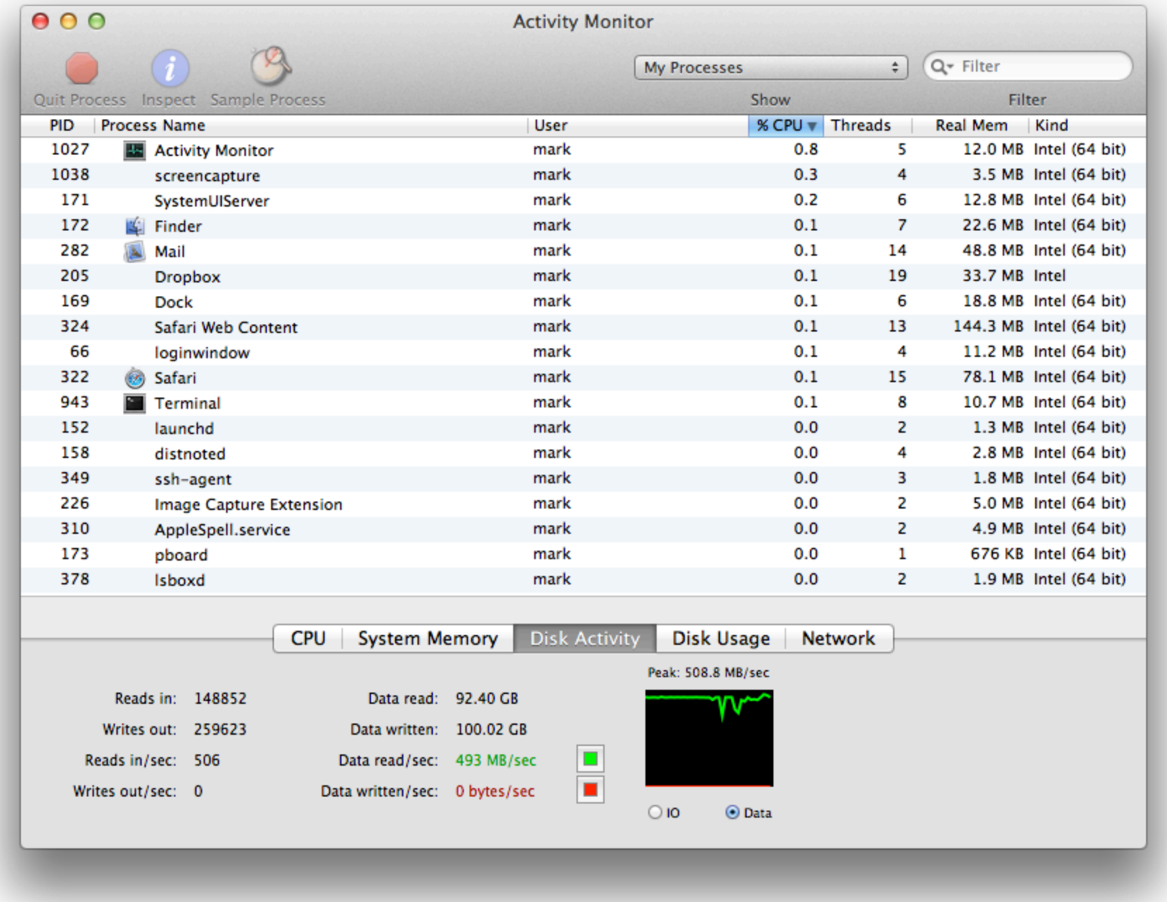
Using a large (100GB) file averages things out over 3-4 minutes to ensure results are consistent and caching effects are mitigated, since my iMac has rather a lot of RAM installed. If you are unfamiliar with the Unix commands, you can start a Terminal and type "`man dd`" to see the manual pages for that command, or "`man purge`" to see the manual page for that command.

Here is a script example of part of the testing:

```
bash-3.2$ df -lh
Filesystem     Size   Used  Avail Capacity  Mounted on
/dev/disk0s2  931Gi  157Gi  774Gi    17%    /
/dev/disk1s2  3.6Ti  1.0Gi  3.6Ti     1%    /Volumes/5DriveRAID5
bash-3.2$ pwd
/volumes/5DriveRAID5
bash-3.2$ dd if=/dev/zero of=junk bs=1024k count=100000
100000+0 records in
100000+0 records out
104857600000 bytes transferred in 176.827257 secs (592994552
bytes/sec)
bash-3.2$ purge
bash-3.2$ dd if=junk of=/dev/null bs=1024k
100000+0 records in
100000+0 records out
104857600000 bytes transferred in 200.161851 secs (523864061
bytes/sec)
```

The performance numbers achieved are visible in parentheses above, the first one listed being for writing to the disk and the second for reading from it.

In order to confirm my results, I ran OS X's built in Activity Monitor (to be found in Applications/Utilities) and monitored the "Data read/sec" (in green) and "Data written/sec" (in red) fields at the bottom of the screen capture below:



PID	Process Name	User	% CPU ▼	Threads	Real Mem	Kind
1027	Activity Monitor	mark	0.8	5	12.0 MB	Intel (64 bit)
1038	screencapture	mark	0.3	4	3.5 MB	Intel (64 bit)
171	SystemUIServer	mark	0.2	6	12.8 MB	Intel (64 bit)
172	Finder	mark	0.1	7	22.6 MB	Intel (64 bit)
282	Mail	mark	0.1	14	48.8 MB	Intel (64 bit)
205	Dropbox	mark	0.1	19	33.7 MB	Intel
169	Dock	mark	0.1	6	18.8 MB	Intel (64 bit)
324	Safari Web Content	mark	0.1	13	144.3 MB	Intel (64 bit)
66	loginwindow	mark	0.1	4	11.2 MB	Intel (64 bit)
322	Safari	mark	0.1	15	78.1 MB	Intel (64 bit)
943	Terminal	mark	0.1	8	10.7 MB	Intel (64 bit)
152	launchd	mark	0.0	2	1.3 MB	Intel (64 bit)
158	distnoted	mark	0.0	4	2.8 MB	Intel (64 bit)
349	ssh-agent	mark	0.0	3	1.8 MB	Intel (64 bit)
226	Image Capture Extension	mark	0.0	2	5.0 MB	Intel (64 bit)
310	AppleSpell.service	mark	0.0	2	4.9 MB	Intel (64 bit)
173	pboard	mark	0.0	1	676 KB	Intel (64 bit)
378	lsboxd	mark	0.0	2	1.9 MB	Intel (64 bit)

Reads in:	148852	Data read:	92.40 GB	
Writes out:	259623	Data written:	100.02 GB	
Reads in/sec:	506	Data read/sec:	493 MB/sec	🟩
Writes out/sec:	0	Data written/sec:	0 bytes/sec	🟥

Results

Here is a table showing the results of the various combinations I found interesting - it is not an exhaustive test of all possible configurations the Pegasus is capable of.

	RAID0	RAID1E	RAID5	RAID6	RAID50
2 Drives	2TB - 100% Write: 368 Read: 365				
3 Drives	3TB - 100% Write: 526 Read: 480		2TB - 66% Write: 351 Read: 358		
4 Drives	4TB - 100% Write: 624 Read: 507				
5 Drives			4TB - 80% Write: 592 Read: 523		
6 Drives	6TB - 100% Write: 626 Read: 504	3TB - 50% Write: 199 Read: 477	5TB - 83% Write: 600 Read: 504	4TB - 66% Write: 539 Read: 517	4TB - 66% Write: 568 Read: 510

Note: Speeds are shown in millions of bytes/second, i.e. MiB/s.

How to interpret the results table above:

Find the number of drives in your array down the left column and the RAID level you want to use across the top. Find the cell where those two intersect and read out the three pieces of information:

a. The available capacity and its percentage space efficiency
b. The sustained write rate in MiBytes/s
c. The sustained read rate in MiBytes/s

Analysis

Looking at a 2 drive RAID0 configuration, it seems to show that you can get around 180 MiB/s from each spindle. If you look at a 3 drive RAID5, you can improve the previous statement and say that you get around 180 MiB/s from each spindle *that is delivering useful data* (i.e. not parity data).
Looking further down the table, you can see that reads saturate the Pegasus somewhere internally (probably the controller) at around 500 MiB/s and that writes saturate it somewhere around 600-630 MiB/s.

Being fairly inquisitive, I wondered whether the Pegasus can deliver twice the performance when two RAID sets are running - and whether there is a way to get even more speed out of it than the maximum 630 MiB/s when writing and 500 MiB/s when reading. The two obvious possibilities were:

1. Create 2 separate 3 Drive RAID0 arrays on the Pegasus and stripe them together within OS X's software RAID in the hope of achieving write performance of 1,052 MiB/s (2x 526 MiB/s) and a read performance of 960 MiB/s (2x 480MiB/s)
2. Create a 4 Drive RAID0 array and a 2 Drive RAID0 array on the Pegasus and stripe them together within OS X's software RAID in the hope of achieving a write performance of 990 MiB/s (368 + 624) and a read performance of 872 MiB/s (365 + 507). Yes, this configuration wastes 2TB of space because the stripes are different sizes - I know that, but was purely interested in the maximum sustained throughput at whatever cost.

Effectively, both of these configurations are 6 Drive RAID0 setups, but the striping is being done partially in the Pegasus and partially in OS X. Well, in the event the following performances were the very best I could achieve:

Configuration	Performance
2 way OS X software stripe across a pair of 3 Drive RAID0 arrays on the Pegasus	Write: 636 Read: 563
2 way OS X software stripe across a 4 Drive RAID0 and a 2 Drive RAID0 array on the Pegasus	Write: 638 Read: 542

So, you can get marginally more performance doing something crazy like this, but you would have to be very desperate for those last few percent of performance to do something like this rather than take the obvious 5 or 6 Drive RAID5 or RAID 6.

Conclusion

I note that the performance difference between 4 Drive RAID 0 and 6 Drive RAID 0 is negligible, or in other words, adding the 2 extra disk spindles going from 4 drives to 6 drives, doesn't increase the throughput, so I suspect something (the controller?) is reaching saturation inside the array. This would appear to be confirmed by the fact that no combination seems to be able to exceed 626 MiB/s write and 523 MiB/s read.

I also noted that, as with nearly all disks, the performance is significantly better at the outer edge than towards the centre of the disk (nearer the spindle). This is because the inner and outer parts of the disk are turning at the same number of revolutions per second, but at the outer edge a greater area of disk surface passes underneath the head in the same time because the radius is greater there. In order to test this, I made a 5TB array and made the first partition 100GB in Disk Utility, followed by a 4,800GB partition, then a 100GB partition at the centre of the disk. The drive sustained 550+MB/s in the first 100Gb partition but barely 300MB/s in the final 100Gb partition. So, if you are using this drive and want to place things in the very fastest area, place it at the outer edge of the disk - or the first partition in Disk Utility!

In summary, the Pegasus Thunderbolt RAID is blazingly fast - and the match of many SSDs, at least in sustained sequential throughput - though probably not in terms of random I/O. As regards the best configuration, well, that will depend on your intended usage - whether for databases, video/photo editing or whatever, but it is easy to see why Promise Technology ships the R6 configured as a 6 Drive RAID 5 because that is almost as fast as any other RAID level, offers a high space efficiency and a good amount of resilience to disk failure.