ZFS L2ARC, ZIL, & SSD

CISC361 - Andrew Chester
What is L2ARC?

L2ARC is a new layer between Disk and the cache (ARC) in main memory for ZFS.

What does it do?

This extra layer provides another cache. When the data being requested is in ARC it can be accessed in sub-microsecond latency. But without L2ARC if its not that means it needs to be read from disk which has a latency in milliseconds.
ZFS L2ARC (cont.)

How does it do this?
Primarily it takes advantage of SSD (but it is possible to use with faster 15K RPM Drives as well) to create another layer of cache that is slower than main memory but faster than disk.

But is it really needed?
It all depends on the situation, if you have a system where a lot of data is being read frequently and you want better performance, then yes. But if not it still won’t hurt performance, only improve it.
But how does it really work?

Well it's pretty simple to understand. When a request for data comes in it is satisfied in the following order:

1. ARC (Adaptive Replacement Cache)
2. vdev cache of L2ARC devices
3. L2ARC
4. vdev cache of Disks
5. Disks

If the data is not in the first layer, it is looked for in the second, then third, and so forth.
The L2ARC populates its cache by periodically reading data from the tail of the ARC. The data in the tail of the ARC is the data that hasn’t been used for while.

If the data is referenced (hit) after the L2ARC has cached it then sure that part of the L2ARC / the time it took to cache it has been wasted, but this rarely happens.

It's also important to note the ARC will never wait on the L2ARC. This prevents the L2ARC from becoming a bottle neck should it become too overworked or begin to fail.
Another nice feature is that L2ARC will never contain dirty data. This means it never has to be flushed to disk, and if the data on disk that it is caching is updated it is dropped from the L2ARC.
What is ZIL?

Well its the ZFS Intent Log! It provides a log of system calls (primarily data being written to disk).

Why?

Well ZFS like other modern filesystems don’t write the data to disk instantly, this would be horrible for performance. But many applications such as a database require assurance that the data has been written to disk. So this is where the ZIL steps in.
Wait but isn’t the ZIL on Disks as well?

Yes ZFS normally uses the pool for the ZIL. But its possible to place the ZIL onto a second set of drives, say SSD drives that have a much higher write performance. This allows writes to be cached and then flushed together to the disks for higher performance.

But what if the power fails?

When the system boots up the data in the ZIL is flushed to disk, this allows your drives to maintain a consistent state.
What is SSD?

Well they’re becoming pretty common and popular so chances are you already have a general idea. SSD without getting too deep into the hardware are essentially really fast USB Flash Drives.

There are two different types of SSD: MLC, and SLC.

SLC, or Single-Level Cell are high performance SSD drives with more write/erase cycles than MLC. Because of this their mean time to failure is better than that of the MLC models. However they cost a lot more.

MLC, or Multi-Level Cell are lower performance SSD (still better than your standard HDD!) that are more dense than the SLC models. The cost of a MLC drive is a lot less than a similar sized SLC but they don’t last as long.
What do we need these for?

Well two examples would be L2ARC and ZIL. SSD disks are necessary when normal drive performance isn’t enough but volatile storage (RAM) can’t be used. (If your ZIL was in RAM and you lost power you could lose the data you were planning to write to Disk.. There goes that database entry of the Check you just deposited!)
Real Life Data!

ECE/CIS

At ECE/CIS we have a Sun X4540 which has 2x Quad Core AMD Opteron processors with 32GB of ram, and up to 48x 1TB HDDs! The machine also has the ability to boot from compact flash and is connected to the network through 10GB Ethernet.

This server is the main research NFS, ISCSI, SMB, AFP, server which serves out all of the research home directories and pools to research groups for storage.
Data (cont.)
So where's the data?

The following times aren’t conclusive but they do offer some insight into the performance. (Un-taring MySQL over NFS)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>No dedicated ZIL</td>
<td>4:26.005</td>
</tr>
<tr>
<td>Dedicated ZIL 2x 18GB Mirrored Log</td>
<td>3:53.645</td>
</tr>
<tr>
<td>No ZIL (Not a good idea)</td>
<td>21.213</td>
</tr>
<tr>
<td>SSD ZIL 2x32GB Mirrored SSD Log</td>
<td>3:38.103</td>
</tr>
</tbody>
</table>
As you can see having a dedicated ZIL increased performance and a SSD ZIL increased performance even more. When the SSD tests were performed other ISCSI tests were being done and may of affected the performance. Also we noticed that the 10GB Ethernet isn’t functioning at the speed it should be which also may be decreasing the performance. Once the issues with the networking have been worked out and the system is back to a normal load the speed should be closer to that of the ZIL turned off (but not as fast).
Sources

http://www.cuddletech.com/blog/pivot/entry.php?id=974

http://blogs.sun.com/brendan/entry/test

http://src.opensolaris.org/source/xref/onnv/onnv-gate/usr/src/uts/common/fs/zfs/arc.c#3579

http://blogs.sun.com/perrin/entry/the_lumberjack

http://hub.opensolaris.org/bin/view/Community+Group+zfs/source