

Breakout 1.1 – Burst Buffer and Novel Scaling

8/4/10

Conversation was mostly about Burst Buffer

- Should/will SSD be in node, in rack, or both?
 - In node may cause jitter for bleed off time
- Should/will SSD be managed by, library/resource mgr, file system, other
 - Industry didn't mind managing SSD if done at IO node if they can own that stack, not happy with managing if in compute node client – cant own that stack entirely
 - Does letting the file system manage the SSD introduce risk beyond just managing in a more simple way
 - Ceph, Panfs, and GPFS all have concepts for stg pools
 - Panfs has the concept of topology awareness as does Ceph so this might not be that difficult to adapt
 - Important concept – could use VM to hold the SSD manager and put it where ever it makes sense.
 - Maybe no real OS on the compute node – complicates life greatly
- Will we need a JCL like language for bleed off and stage in?
 - Very likely
- Will we need to store distribution information so that stage in can be intelligent as to which SSD to load which data to, or just load to data to any SSD and sort it out over the interconnect
 - no consensus
- How global should the SSD be, trade off between inconvenience for stage in resorting and large failure domain
 - Probably keeping failure domain small makes sense
- Reliability for burst buffer SSD –
 - it wont see that many writes (once per 3 hours) so like 10-15 years endurance, keep failure domain small etc.

More Burst Buffer Stuff

- Should the SSD be connected to the interconnect or not
 - Not necessary for speed – just SAS may keep up with speed necessary since nodes will be so memory poor/core
 - SAS parts are fast and extremely cheap, even switches for multi-pathing
 - Topology awareness would be needed, and the more complex the interconnect technology the uglier this gets
 - Connection to the interconnect could call Jitter
- Should the checkpoint flow through the IO node (from compute node through node to SSD)
 - Not necessary, but doable
- Getting awfully close to the third rail – Active disk
 - Since the burst buffer is SSD, can't you do fancy stuff like scatter gather to/from memory to/from disk, dedup, etc?
- Globally accessible (the name or the entire file)
 - Once the checkpoint is written, can you make the name available in the name space on the file system?
 - Do you want to do this, do you want to bother the SSD from things outside the large machine
 - Can this help viz/data analysis of data before its written to the disk file system
- Won't you make the node way more costly to put flash in it
 - Well you don't need to put it in the node but if you did, it could just be a chip on the mother board
 - It would not necessarily require SAS, it could be anything
 - You don't need this to be that fast – nodes will be memory poor