Matthew

Monday, October 19, 2015 10:26 AM

Static organization is not good
Need to optimize where data is going based on changing system condition

Different types of I/O from the NNSA perspective

POSIX is global shared memory abstraction
Can have false sharing
Locks are coarse
Consistency semantics are strong

Attributes suck in keeping metadata alive

Lazy attributes would speed performance

Sirroco is a clean sheet redesign
Sirocco object store (SOS)
Low level object based system
System level api
It is storage not a file system
Hierarchical fixed depth
Storage and security only

On top of this are smart clients that know how to use the object system
Posix or hdf
Etc

Complicated data management would be hidden by the smart client
Object sotrage point of view they are client
From the user point of view they are the service

Construction sirroco is based on the LWFS philosophy

HDF is a data model

You provide the very basics of storage
Clients bring their own services
And use whatever is available
Not human friendly names in sir

Clients can work with object ids

Inspired by unstructured p2p systems

Servers can come and go at anytime
Clients can come and go at anytime
Clients are leeches
This is fully elastic
Can you take out portion of the service at anytime

Can servers push information about reliability/performance to the client?
- Clients can be informed about busyness
- Servers can exit/leave during writes
- Methods to maintain consistency

You can explicitly define location or do a p2p overlay