**OFI WG Data Storage / Data Access Subteam Weekly telecom – 11/04/2014**

**OFIWG Download Site:** [www.openfabrics.org](http://www.openfabrics.org) 🡪OFED/OFA Resources 🡪 OpenFabrics Interfaces WG

**Agenda**

* role call,
* Object-based storage, cloud applications

Paul introduced the notion of object I/O, and described it for our purposes as being defined by the API not by the storage paradigm. For example, even though Lustre is described as an object storage system, it is still using a POSIX interface to do file reads and writes, where the files are stored by the OSTs are objects. He mentioned a RESTful API as one example of object I/O. Bart mentioned CEPH is another example.

Linden – thinks we need this level of abstraction. Some object systems are based on ‘best effort’, but would like to see it pointed more toward high performance. Many RESTful interfaces today are used for HTTP transfers e.g. access to the cloud. For high performance, would like to be able to access very large objects without breaking it up into little bitty transfers, or even avoid transferring the whole object, e.g. support for accessing data at some offset into the object. Combined with GET/PUT kind of statements, it looks more and more like simple data accesses: a pointer and an extent.

(??) Another aspect is the ability to get various revisions of the object. This may imply computations at the target to find an earlier version of the object in question. For example, the distribution of the versions may imply some kind of a scatter/gather operation.

Sean – hearing similar requests to do versioning in the target - which may require re-calculating addresses or offsets of SGLs.

(??) Also consider compression.

Linden – Expand that to include encryption.

Many environments are based on key/value and don’t allow re-writes. For example, when storing a new version, do you store using the same key/value, or is a new object being stored, invisible to the storage system. In other words, does this happen at a level above us allowing us to keep the storage interface completely clean by separating out the management of objects from the storage of those objects. Not clear where this group should land in that continuum. First thought – don’t get in the middle of that and stick with a simple get/put architecture.

**Agenda for next meeting**

Requirements discussion for byte-addressable memory – Bernard Metzler

**Next regular telecom**

Next meeting: Tuesday, 11/18/14

8am-9am Pacific daylight time

**NOTE:** We have switched over to using Webex (courtesy of Cisco). The URL for joining meetings is:

<https://cisco.webex.com/cisco/j.php?J=200935598&PW=67935ad6df07030d5f05044a5b0f>