An Introduction to the OpenFabrics Interface

#OFAUserGroup
Paul Grun – Cray
w/ slides stolen (with pride) from Sean Hefty
Agenda

• Where the OFA is going
• Forming the OFI WG
• First Principles
  – Application-centric I/O
  – Transport Agnostic
• Architecture overview
• Release process, timeline
• When should you get involved?
Objective: accelerate adoption of IB and related technologies. The OFA was mainly an enabling body.
An interesting development

- IB Spec
- iWARP Spec
- libibverbs
- kernel
- vendors
- distros
- users
An interesting development

For the 1st time, the OFA is developing new technology, without an underlying industry standard specification.

Pure, requirements-driven, technology development.
An interesting development

We recently added the OFW WG as a new technical working group
An interesting development

Technical Advisory Council
- OFA’s technology incubator
- observe trends in technology
- clearing house for new ideas
- spawn new development efforts

The OFA has really become a technology development organization
Original Objectives for OFI

- Maximize application I/O (aka network) effectiveness
- Excellent support for a wide range of (classes of) applications
- Minimize interface complexity and overhead
- Make the interface(s) extensible
- Not constrained to a particular wire, fabric or vendor

March 30 – April 2, 2014  #OFADevWorkshop

Pretty much on track!

Maybe not so much
OFA Board Ask

Create an OpenFramework (OFWG) working group to:

- Develop, test, and distribute:
  1. An extensible, open source framework that provides access to high-performance fabric interfaces and services.
  2. Extensible, open source interfaces aligned with ULP and application needs for high-performance fabric services.

OFWG will not create specifications, but will work with standards bodies to create interoperability as needed.
1st Principle

What are the central requirements of consumers of a network API?

“Application-centric I/O” is the art and science of defining an I/O architecture to maximize application effectiveness

Consumer orientation has emerged as a key watchword for the OFA
 Classes of Applications

OFI APIs are being driven by requirements from specific classes of applications

<table>
<thead>
<tr>
<th>Legacy apps (skts, IP)</th>
<th>Data Analysis</th>
<th>Data Storage, Data Access</th>
<th>Distributed and Parallel Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Skts apps</td>
<td>- Structured data</td>
<td>- Filesystems</td>
<td>Via msg passing</td>
</tr>
<tr>
<td>- IP apps</td>
<td>- Unstructured data</td>
<td>- Object storage</td>
<td>- MPI applications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Block storage</td>
<td>- Distributed storage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Distributed storage</td>
<td>- Storage at a distance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Via shared memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- PGAS languages</td>
</tr>
</tbody>
</table>
Classes of Applications

OFI APIs are being driven by requirements from specific classes of applications.

<table>
<thead>
<tr>
<th>Legacy apps (skts, IP)</th>
<th>Data Analysis</th>
<th>Data Storage, Data Access</th>
<th>Distributed and Parallel Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Skts apps</td>
<td>- Structured data</td>
<td>- Filesystems</td>
<td>Via msg passing</td>
</tr>
<tr>
<td>- IP apps</td>
<td>- Unstructured data</td>
<td>- Object storage</td>
<td>- MPI applications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Block storage</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Distributed storage</td>
<td>- PGAS languages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Storage at a distance</td>
<td></td>
</tr>
</tbody>
</table>

DS/DA WG

OFI WG
**2nd Principle**

libfabrics

Application requirements

Fabric Interfaces

- Control Interface
- Message Queue
- RDMA
- Atomics
- CM Services
- Active Messaging
- Tag Matching
- Collective Operations

Fabric Provider Implementation

- I/O service
- I/O service
- I/O service

Multiple Providers

APIs (driven by OFA ‘interest groups’)
Development

Rough conceptual model → Requirement analysis

Quarterly release cycle

Deployment

Iterative design and implementation

Input from wide variety of devices

Collective feedback from OFIWG

~200 requirements MPI, PGAS, SHMEM, DBMS, sockets, …
Design

- Enable simple, basic usage
- Move functionality under OFI
- Advanced application constructs
- Expose abstract HW capabilities

Implementation

Agnostic

EASY

GURU

Range of usage models
Architecture

Open Fabrics Interfaces (OFI)

- Control Services:
  - Discovery
  - fi_info

- Communication Services:
  - Connection Management
  - Address Vectors

- Completion Services:
  - Event Queues
  - Counters

- Data Transfer Services:
  - Message Queues
  - Tag Matching
  - RMA
  - Atomics

OFI Enabled Applications

MPI  SHMEM  PGAS
Fabric Information

Endpoint Types

- MSG
  - Reliable connected
- DGRAM
  - Datagram
- RDM
  - Reliable datagram messages
  - Reliable unconnected

Capabilities

- Message queue
  - FIFO
- RMA
- Tagged messages
  - Sends match with specific receive buffers
- Atomics

Select desired endpoint type and capabilities
• Capabilities
  – Application desired *features* and *permissions*
  – Primary capabilities
    • Must be requested by application
  – Secondary capabilities
    • May be requested by application
    • May be offered by provider

• Attributes
  – Defines the *limits* and *behavior* of selected interfaces
  – Negotiated

• Mode
  – Provider request on application
Providers

Under development for Release 1.0
- sockets
- PSM
- verbs
- Usnic
- others expected
OFI 1.0 Providers

• **Sockets**
  – Implement all interfaces and functionality
  – App. development & debug

• **Verbs**
  – Targets any verbs HW
    • Not optimized for a specific device
  – Only common verbs functionality supported

• **PSM**
  – Targets non-verbs HW
  – Expands capabilities beyond lower software driver

• **USNIC**
  – Targets non-verbs HW
  – Cisco will address

Input from verbs derivative and non-verbs providers also fed into OFI design
Release timeline

• time-based release process
• quarterly releases planned for the early stages
• Release 1.0 rc2 available now
• Release 1.0 rc3 planned for end of March
• Release 1.0 soon thereafter
When to get involved

• Walking a fine line:
  – Don’t want to release too early, but…
  – Need to get the broader community involved

The time to get advanced developers involved is now!
“If Verbs Programming is like the assembly language version of network programming, OFI is the C language version.”

Doug Ledford
Thank You