Evolving OFS – Teaching sharks to swim, from the top down

#OFADevWorkshop
What was said on Monday…

Technology is like a shark…it’s always moving.

It if stands still, it dies.

This session is about how to keep the state of the art in I/O moving.
Objectives:
1. OFA remains the premier provider of I/O software for HPC
2. OFA becomes the premier provider of I/O software for Enterprise and other segments including Big Data, Clouds

To accomplish these objectives, OFS must…
…be the most scalable I/O solution on the planet
…deliver I/O services that are attractive to its users
…incorporate the latest technologies, such as multi-core, NVM, others
The point behind application-centric I/O is to allow applications to communicate with each other, in an interoperable way.
The method for accomplishing this is by allowing applications to directly access each other’s memory.

They do this by using industry wire protocols.
Application-centric I/O

Its name comes from the fact that the entire I/O infrastructure is derived by looking at the requirements of the application.
Elements of app-centric I/O

Goal: applications communicate directly
Method: message passing
a standard i/f to a message passing service
standard wire protocols transport messages
**RDMA** is the ability to remotely access another user’s virtual memory space.

RDMA is the playground within which OFA plays.
RDMA, IB

This is the goal – “RDMA”

- remote memory access
  - via message passing
  - Verbs as an interface
  - IB specific mechanisms
  - Wire level protocols

This is the mechanism to achieve the goal – “InfiniBand”
RDMA, IB

This is what OFA cares about

remote memory access
via message passing
Verbs as an interface
IB specific mechanisms
Wire level protocols

This is what we live with
The role of standards

The (IB/RoCE/iWARP) standard governs the mechanism for transporting messages, and the format and layout of the packets comprising the message as they appear on the wire.

If the layout of the headers changes, the standard has to change too.
This was the idealized vision for OpenIB. It took advantage of the verbs interface defined in the IB spec to allow applications to pass messages. The messages were carried in packets according to a standardized wire protocol.

Remote memory access via message passing

Verbs as an interface

IB specific mechanisms

Wire level protocols

This is what OFA cares about
Today’s reality

- “ideal” applications
- legacy apps
- MPI
- PGAS

ULPs

ULP

remote memory access
via message passing

Verbs as an interface
IB specific mechanisms
Wire level protocols

Coded to the verbs API
Today’s reality

Our goal now is to **optimize and extend** this architecture. Teaching a shark to swim, so to speak.

- “ideal” applications
- legacy apps
- MPI
- PGAS

ULPs

ULP

remote memory access
via message passing

Verbs as an interface
IB specific mechanisms
Wire level protocols
Optimizing MPI

Verbs was thought to a poor match with MPI

MPI

remote memory access
via message passing

Verbs as an interface
IB specific mechanisms
Wire level protocols
Optimizing MPI

PSM addressed this mismatch, but in a vendor specific way.

**Request:** Solve this problem in a vendor neutral way

**Owner:** OFA

- remote memory access via message passing
- Verbs as an interface
- IB specific mechanisms
- Wire level protocols

This i/f is not specified
**Optimizing PGAS**

**Request**: innovate here by consulting with compiler writers.

**Request**: define one (or more) conduit(s) based on the chosen interface.

**Owner (for both)**: OFA

**Request** (if needed): optimize the provider. But only if changes are required in the wire protocol or the verbs specification.

**Owner**: network standards body

---

**PGAS**

- **GASnet**
- **conduit(s)**
- **provider(s)**
- **nic(s)**

**remote memory access**

via message passing

**Verbs as an interface**

**IB specific mechanisms**

**Wire level protocols**
Consult with application class users to innovate at this level

Where practical, write ULPs to the verbs i/f

Work with standards bodies to define the provider and NIC as needed

remote memory access
via message passing

Verbs as an interface
IB specific mechanisms
Wire level protocols

application class

ULP

provider(s)

nic(s)
3 categories of ownership

Ownership: OFA

Ownership: Joint

Ownership: Standards organization

application class

ULP

provider(s)

nic(s)

remote memory access via message passing

Verbs as an interface

IB specific mechanisms

Wire level protocols
Some requests (examples)

**Request:** Add support for Big Data, other classes of apps

**Owner:** OFA

**ILLUSTRATIVE EXAMPLE ONLY**
Some requests (examples)

**Request:** Figure out how to deploy NVM.  
**Owner:** OFA

ILLUSTRATIVE EXAMPLE ONLY
Some requests (examples)

Request: Lighten the verbs interface
Owner: OFA, IBTA

ILLUSTRATIVE EXAMPLE ONLY
Some requests (examples)

Request: reduce memory footprint by creating “reliable connectionless” service
Owner: IBTA

ILLUSTRATIVE EXAMPLE ONLY
Some requests (examples)

**Request**: Reduce cost of memory registration

**Owner**: IBTA (maybe)

---

- Application class
- ULP
- Remote memory access via message passing
  - Verbs as an interface
  - IB specific mechanisms
  - Wire level protocols

ILLUSTRATIVE EXAMPLE ONLY
Some requests (examples)

Request: add channel bonding
Owner: IBTA
(request made by Whamcloud/OpenSFS)

ILLUSTRATIVE EXAMPLE ONLY
Proposal:

- OFA’s Technical Advisory Council to own the process for the OFA.

- Obviously, code is not created “by the OFA”. Code is created and contributed by OFA members, all under the usual licensing rules.
Discussion?
• I believe we can usefully limit the mission of the OFA as being the development and promotion of RDMA-related SW stacks, *provided* we use the broadest possible definition of RDMA

• I believe there is RDMA, my known universe, and everything else that is not RDMA, which I don’t think I need to care about

• I believe Paul has given us a useful model for examining challenges, opportunities and so on, and I commit to using it. As Chair, I charge the TAC with taking this on
  • If this model identifies challenges that need to be responded to that don’t fit with RDMA, we need to figure it out. That doesn’t mean we wish it away; it means we figure it out
Jim’s conclusions on behalf of the OFA (cont’d)

• I believe OFA needs to take a leadership position WRT RDMA and “the other” in three areas:
  • What the OFA can do on its own, i.e., develop “app centric APIs”
  • What the OFA needs to do in the way of crafting “asks” of other orgs, e.g., possible spec enhancements in response to problems and concerns reported to us
  • What the OFA may need to do in “other areas”, which remain to be identified and understood
• We have examples of concerns, for example in scalability, that appear to be straight-forward extensions to the existing IBA specs (for example, because this is most familiar) and others (e.g., adaptive routing) that appear to go beyond it.
  • However, this seems to be a distinction not worth undertaking for the OFA. The better approach is to toss the package over to the spec owners and let them decide
  • Again, we’re going to assume the world is RDMA, so we’ll communicate with spec owners and stakeholders and let them decide
    • Anything they identify as being outside of their purview will be *really* interesting, because it will challenge my “RDMA only” POV
Jim’s conclusions on behalf of the OFA (cont’d)

- Understand that this is a broader range of responsibilities than the OFA has accepted in the past:
  - We’re being encouraged to *lead*
    - API development
    - Proactive requests for spec development
  - Serious consideration of that which may be “beyond RDMA” and whatever that means in terms of future activities
    - This may sound obvious, but we’ve considered high-performance but non-RDMA interconnects in the past, and decided against them

Executing this process successfully requires discipline and a deep understanding of the different roles of the OFA and standards bodies
Thanks!
Elements of app-centric I/O

Goal: applications communicate directly
Method: message passing
a standard i/f to a message passing service
standard wire protocols transport messages

remote memory access
remote memory access method
interface
mechanisms
Wire level protocols
Elements of app-centric I/O

Goal: applications communicate directly
Method: message passing

a standard i/f to a message passing service
standard wire protocols transport messages

remote memory access
message passing
verbs
IB mechanisms
wire level protocols
RDMA is NOT the same thing as InfiniBand

InfiniBand is a collection of mechanisms, and an architecture for implementing RDMA

It specifies the ON-the-WIRE protocols that allow one user to access another’s memory