{over}

XioMessenger

Ceph transport abstraction based on Accelio, a high-performance message-passing framework by Mellanox.

Key benefit: efficient adapter for Infiniband/RDMA transports using Openfabrics interfaces (ibverbs).

Futures: multi-protocol (TCP, memory, others) transport abstraction with advanced multi-path support (among other features)

History

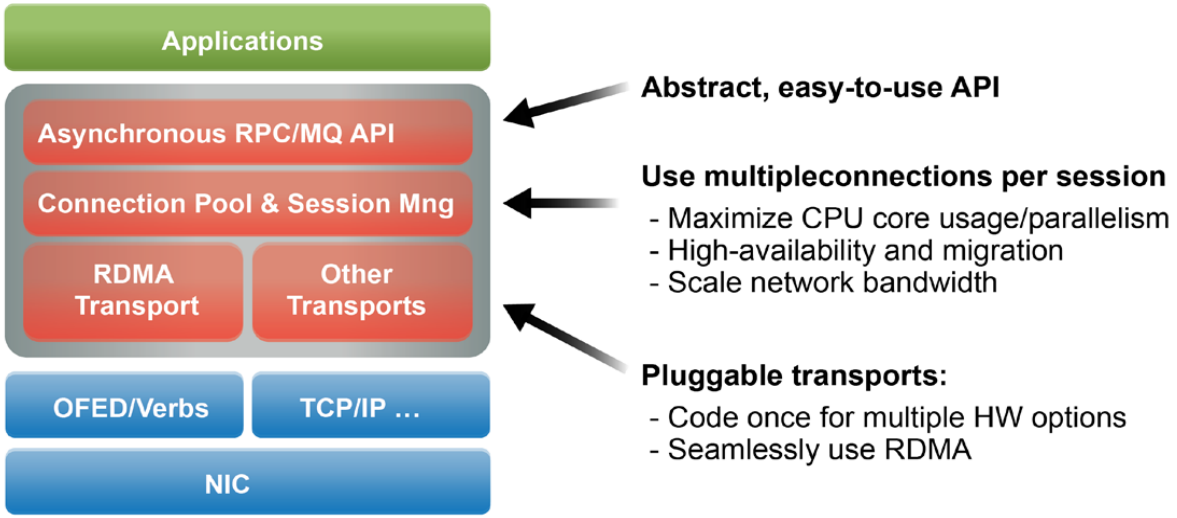
Work funded by Mellanox in support of customers using Ceph.

Objectives

- Increase Ceph transport flexibility
- Support efforts to increase Ceph i/o performance

What is Accelio (from Accelio WP)

- High-performance, asynchronous, reliable messaging library built with hardware acceleration/RDMA in mind
  - framework for building high-performance RPC transports



Some Key Features of Accelio

- streamlined, but flexible selection of messaging models
  - request-response, one-way
- reliable delivery primitives
- zero-copy
- internally (almost) lockless
  - including API-visible, lock-free memory pool allocator
- optimized for thread/CPU parallelism, NUMA, with minimal impact on application complexity

Accelio Roadmap

- v1.1 (Spring 2014) - Ceph XioMessenger Support
- v1.x (Fall 2014)

Notable Fall 2014 Planned Features

- Initial TCP and mixed transport support (pref transport and fallback policies, etc)
- New flow-control interfaces
- More

Accelio Available Performance

Measurement on modern Intel equipment and Mellanox infiniband switch and HBA hardware indicates ability to deliver full bandwidth and latency potential to applications, at modest CPU cost.

- measured >3,000,000 IIOPs (round-trip message latency)
- measured up to 6GB/s message bandwith on single-port FDR (saturation)

Benchmark codes using all delivery models are included with Accelio source for measurement/verification.

Getting Accelio

Source code (dual BSD, GPL license):

https://github.com/accelio/accelio

More Info

http://www.accelio.org/wp-content/themes/pyramid_child/pdf/WP_Accelio_OpenSource_IO_Message_and_RPC_Acceleration_Library.pdf

What is XioMessenger

Accelio adapter for Messenger, the Ceph transport layer abstraction

Objectives

- Drop-in replacement/alternate for TCP SimpleMessenger
- Take full advantage of Accelio capabilities for
  - zero-copy
  - thread/CPU parallelism (portals, minimize locking)

Ceph Messenger Key Abstractions

- Messenger : bi-directional communication endpoint
- Connection : active communication channel with a remote
  - supports ordered delivery
  - in full form supports more advanced flow-control/rate-limiting (not currently supported fully in XioMessenger)
  - in current full form, supports strong endpoint identification and wire encryption using host credential (CephX, not currently supported in XioMessenger)
- Message : typed message (request, reply, or other) in one direction
  - (de)serialized on (receipt)send using Ceph encode/decode and buffer::list primitives

XioMessenger Key Abstractions

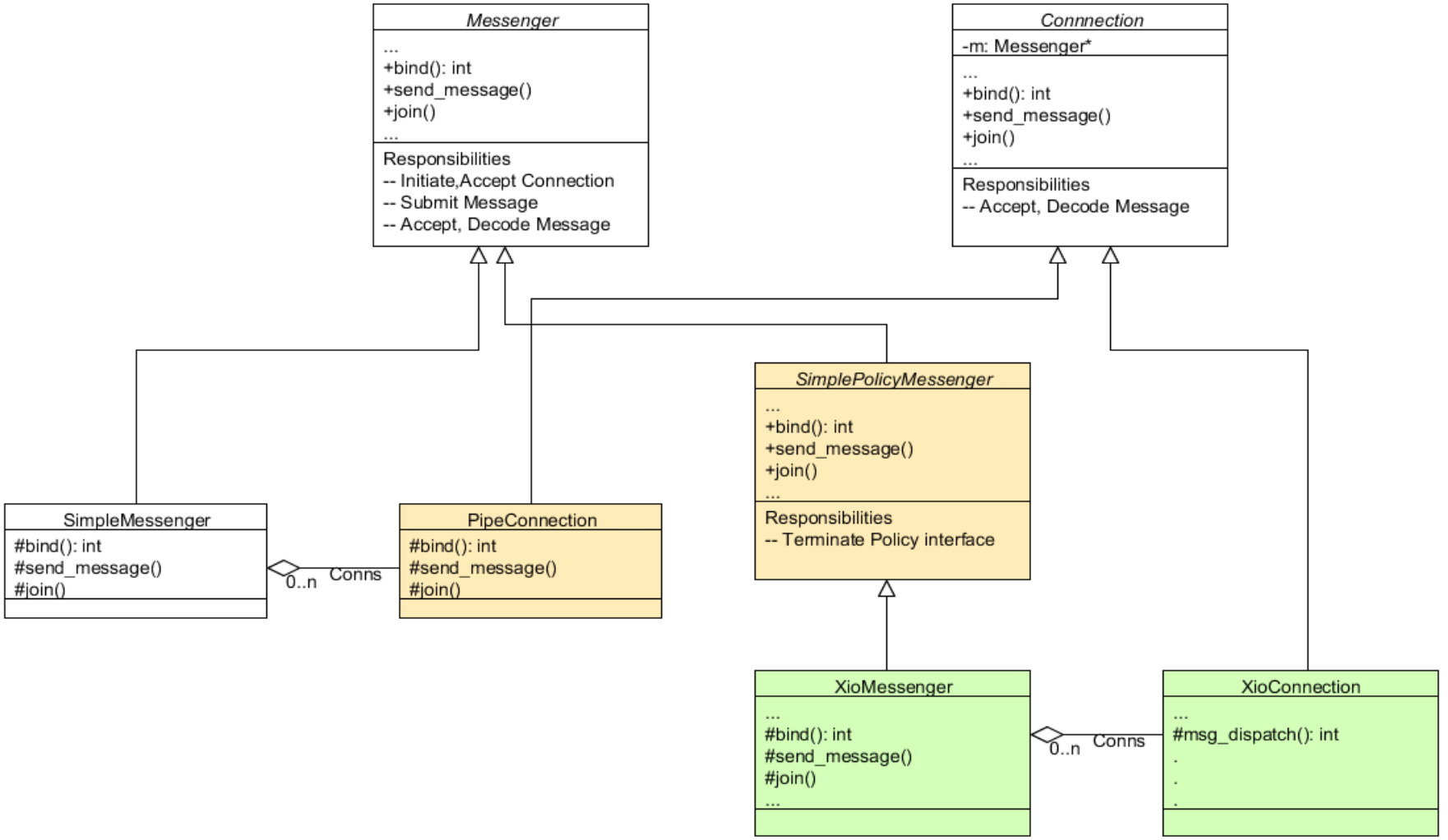
• XioMessenger : implements Messenger interface, encapsulates Accelio network bindings, established connections (Accelio sessions and connections), and a set of thread/event loops (portals, below)

• XioConnection : implements Connection interface, represents an inbound or

• outbound session/connection (currently 1:1 in XioMessenger)

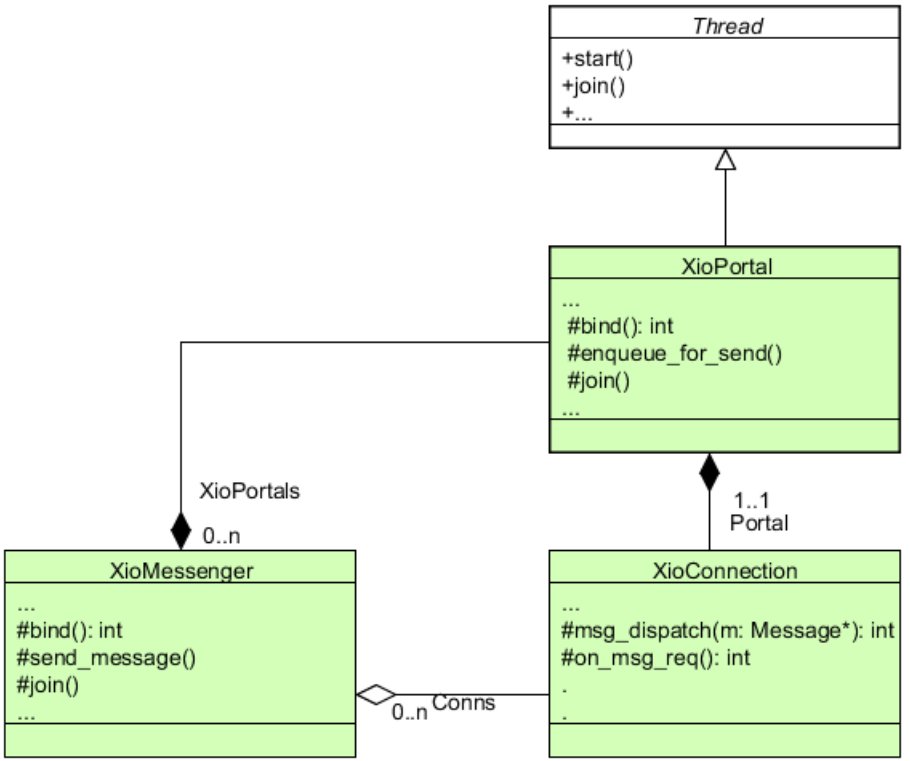
• XioLoopbackConnection : encapsulates a connection from a host to itself, implements direct delivery of Ceph messages to the local host, without serialization

Revised Messenger class hierarchy



XioMessenger Key Abstractions

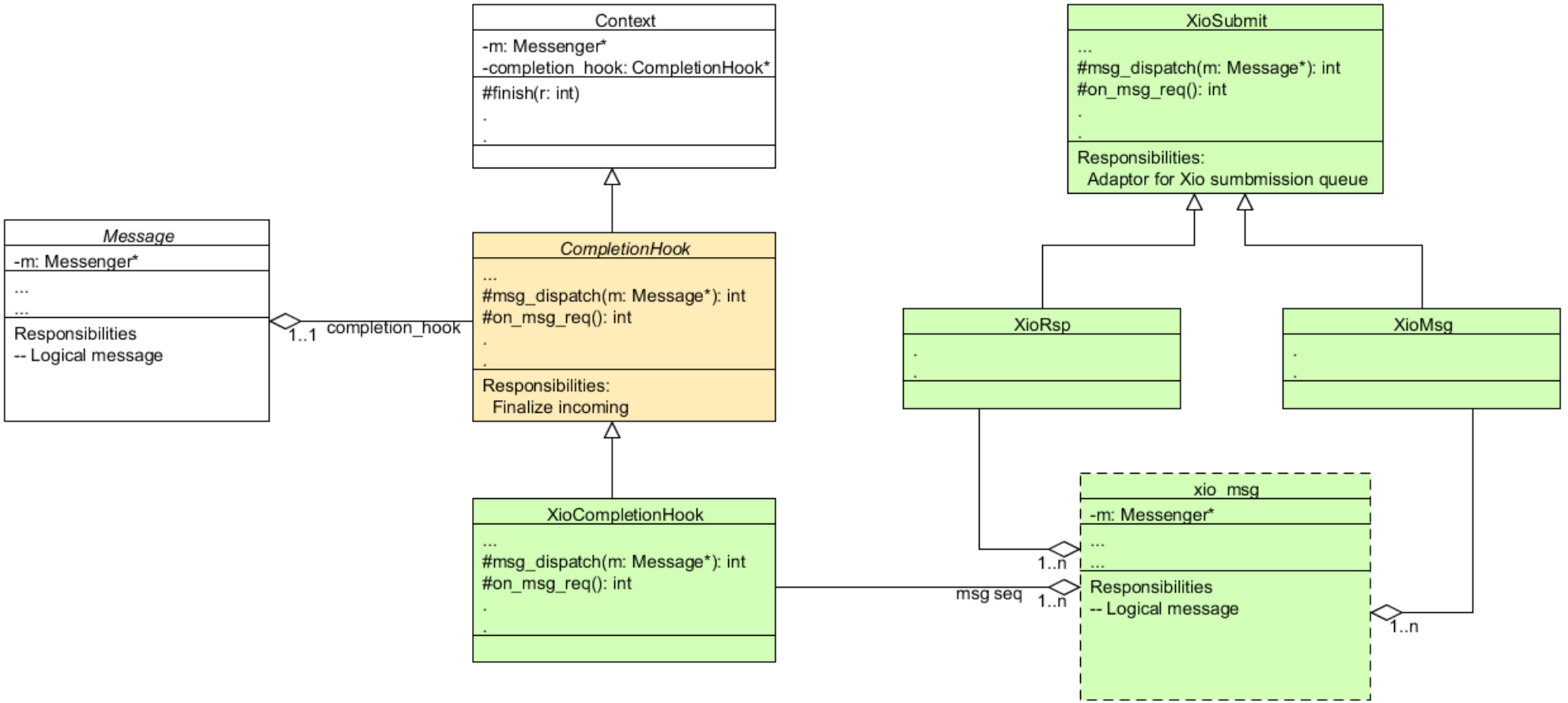
- XioPortal : encapsulates an Accelio thread context, or portal



XioMessenger: Workflow Abstractions

- XioPool : a set of related Accelio mempool memory handles, that can be disposed together
- XioSubmit : abstract interface for Accelio messages
- XioMsg (->XioSubmit) : outgoing one-way message, encapsulates a 1..n xio_msg structures
- XioRsp (->XioSubmit) : outgoing one-way message completion
- XioCompletionHook (->CompletionHook) : Accelio implementation of new Message completion functor

XioMessenger workflow



CohortFS

Mellanox TECHNOLOGIES

XioMessenger Benchmarking

* xio_client/xio_server {src/test/messenger}

parameters

--addr {ip address of RDMA-capable interface}

--port {ip port}

--dsize {data size, in bytes, eg, 4096}

Most Recent Benchmark Results

test date	app	CRC	protocol	HCA	# of hosts	msg depth	buffer size	iiops (1-way ops/s)	bandwidth (MB/s)
03/03/14	simple_messenger	crc	tcp ipoib	n/a	2	50	4K	10672.2251 (21344.4502)	41.6883 (83.3766)
03/03/14	simple_messenger	no_crc	tcp ipoib	n/a	2	50	4K	11136.4872 (22272.9744)	43.5019 (87.0038)
03/03/14	simple_messenger	crc	tcp ipoib	n/a	2	50	64K	3261.6658 (6523.3316)	203.8541 (407.7082)
03/03/14	simple_messenger	no_crc	tcp ipoib	n/a	2	50	64K	4655.6035 (9311.2070)	290.9752 (581.9504)
03/03/14	xio_messenger	no_crc	rdma	ConnectX-3	2	50	0K	512000	39.05
03/03/14	xio_messenger	no_crc	rdma	ConnectX-3	2	50	4K	256000	1000
03/03/14	xio_messenger	no_crc	rdma	ConnectX-3	2	50	64K	51328.3208	3208.02

XioMessenger Implementation Status

- Integration with all Ceph daemons and clients code complete
- RBD testing possible with rados test tool (--xio)
- Finalization of open issues for full cluster testing in progress

Getting Ceph with Accelio

In official Accelio repository, use the for_next branch (includes API extensions prototyped for XioMessenger, and known to work with the published branches).

https://github.com/accelio/accelio

Branches on CohortFS Ceph repository (pre-merge) provided for Autotools and CMake build systems.

https://github.com/linuxbox2/linuxbox-ceph

- automake, xio-firefly
- cmake (what we use), xio-firefly-cmake

Accelio-specific Config Parameters

- rdma local (ip addr) : bind to specific RDMA interface
- cluster_rdma (boolean) : use XioMessenger as default cluster messenger
- client_rdma (boolean) : in clients, use XioMessenger as client messenger

example

```
rdma local = 10.17.23.10
cluster_rdma = true
client_rdma = true
```

XioMessenger Limitations

- no CephX (not yet scoped)
- flow control/recovery semantics (in progress)

XioMessenger Upstreaming Status

- in progress (wip-xio)

Q/A

{over}

6/10/2014 Matt Benjamin matt@cohortfs.com

XioMessenger

Ceph transport abstraction based on Accelio, a high-performance message-passing framework by Mellanox.

Key benefit: efficient adapter for Infiniband/RDMA transports using Openfabrics interfaces (ibverbs).

Futures: multi-protocol (TCP, memory, others) transport abstraction with advanced multi-path support (among other features)

History

Work funded by Mellanox in support of customers using Ceph.

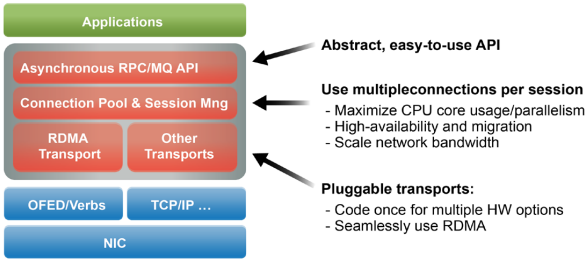
Objectives

- Increase Ceph transport flexibility
- Support efforts to increase Ceph i/o performance

CohortFS

What is Accelio (from Accelio WP)

- High-performance, asynchronous, reliable messaging library built with hardware acceleration/RDMA in mind
  - framework for building high-performance RPC transports



Some Key Features of Accelio

- streamlined, but flexible selection of messaging models
  - request-response, one-way
- reliable delivery primitives
- zero-copy
- internally (almost) lockless
  - including API-visible, lock-free memory pool allocator
- optimized for thread/CPU parallelism, NUMA, with minimal impact on application complexity

Accelio Roadmap

- v1.1 (Spring 2014) - Ceph XioMessenger Support
- v1.x (Fall 2014)

Notable Fall 2014 Planned Features

- Initial TCP and mixed transport support (pref transport and fallback policies, etc)
- New flow-control interfaces
- More

Accelio Available Performance

Measurement on modern Intel equipment and Mellanox infiniband switch and HBA hardware indicates ability to deliver full bandwidth and latency potential to applications, at modest CPU cost.

- measured >3,000,000 IIOPs (round-trip message latency)
- measured up to 6GB/s message bandwith on single-port FDR (saturation)

Benchmark codes using all delivery models are included with Accelio source for measurement/verification.

Getting Accelio

Source code (dual BSD, GPL license):

https://github.com/accelio/accelio

More Info

http://www.accelio.org/wp-content/themes/pyramid_child/pdf/WP_Accelio_OpenSource_IO_Message_and_RPC_Acceleration_Library.pdf

What is XioMessenger

Accelio adapter for Messenger, the Ceph transport layer abstraction

Objectives

- Drop-in replacement/alternate for TCP SimpleMessenger
- Take full advantage of Accelio capabilities for
  - zero-copy
  - thread/CPU parallelism (portals, minimize locking)

Ceph Messenger Key Abstractions

- Messenger : bi-directional communication endpoint
- Connection : active communication channel with a remote
  - supports ordered delivery
  - in full form supports more advanced flow-control/rate-limiting (not currently supported fully in XioMessenger)
  - in current full form, supports strong endpoint identification and wire encryption using host credential (CephX, not currently supported in XioMessenger)
- Message : typed message (request, reply, or other) in one direction
  - (de)serialized on (receipt)send using Ceph encode/decode and buffer::list primitives

XioMessenger Key Abstractions

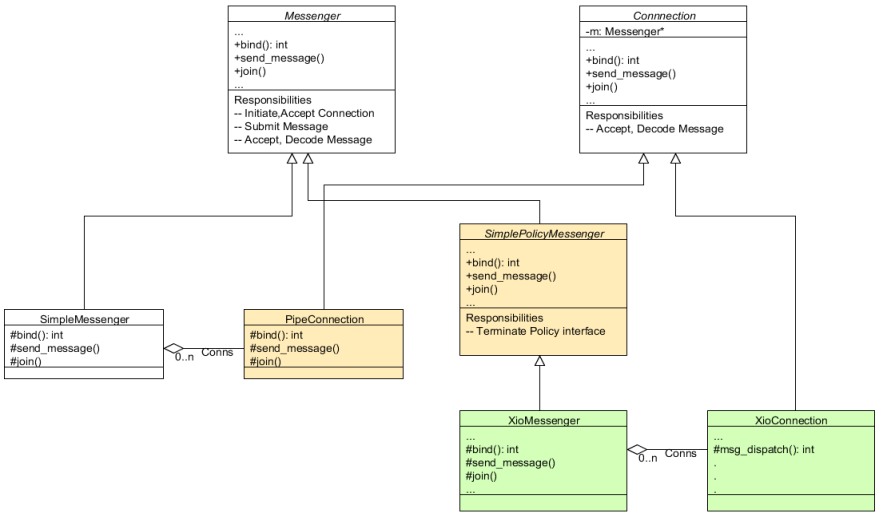
• XioMessenger : implements Messenger interface, encapsulates Accelio network bindings, established connections (Accelio sessions and connections), and a set of thread/event loops (portals, below)

• XioConnection : implements Connection interface, represents an inbound or

• outbound session/connection (currently 1:1 in XioMessenger)

• XioLoopbackConnection : encapsulates a connection from a host to itself, implements direct delivery of Ceph messages to the local host, without serialization

Revised Messenger class hierarchy



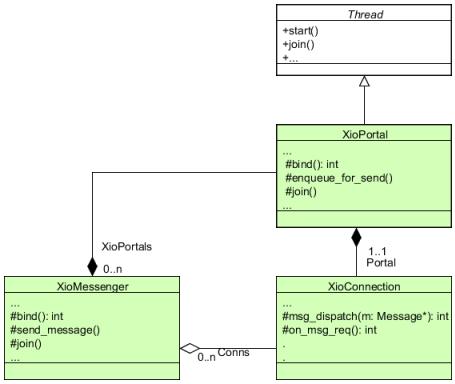
CohortFS

Mellanox TECHNOLOGIES

CohortFS

XioMessenger Key Abstractions

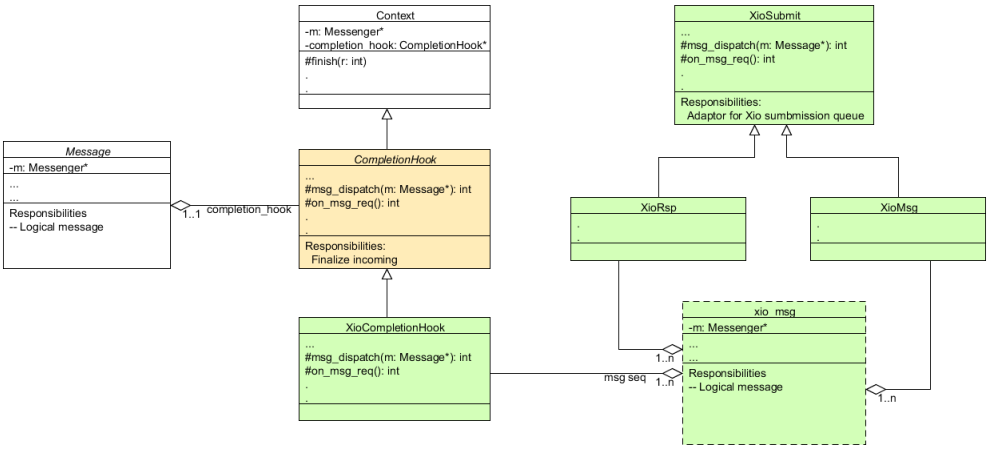
- XioPortal : encapsulates an Accelio thread context, or portal



XioMessenger: Workflow Abstractions

- XioPool : a set of related Accelio mempool memory handles, that can be disposed together
- XioSubmit : abstract interface for Accelio messages
- XioMsg (->XioSubmit) : outgoing one-way message, encapsulates a 1..n xio_msg structures
- XioRsp (->XioSubmit) : outgoing one-way message completion
- XioCompletionHook (->CompletionHook) : Accelio implementation of new Message completion functor

XioMessenger workflow



CohortFS

Mellanox TECHNOLOGIES

XioMessenger Benchmarking

* xio_client/xio_server {src/test/messenger}

parameters

--addr {ip address of RDMA-capable interface}

--port {ip port}

--dsize {data size, in bytes, eg, 4096}

Most Recent Benchmark Results

test date	app	CRC	protocol	HCA	# of hosts	msg depth	buffer size	iiops (1-way ops/s)	bandwidth (MB/s)
03/03/14	simple_messenger	crc	tcp ipoib	n/a	2	50	4K	10672.2251 (21344.4502)	41.6883 (83.3766)
03/03/14	simple_messenger	no_crc	tcp ipoib	n/a	2	50	4K	11136.4872 (22272.9744)	43.5019 (87.0038)
03/03/14	simple_messenger	crc	tcp ipoib	n/a	2	50	64K	3261.6658 (6523.3316)	203.8541 (407.7082)
03/03/14	simple_messenger	no_crc	tcp ipoib	n/a	2	50	64K	4655.6035 (9311.2070)	290.9752 (581.9504)
03/03/14	xio_messenger	no_crc	rdma	ConnectX-3	2	50	0K	512000	39.05
03/03/14	xio_messenger	no_crc	rdma	ConnectX-3	2	50	4K	256000	1000
03/03/14	xio_messenger	no_crc	rdma	ConnectX-3	2	50	64K	51328.3208	3208.02

XioMessenger Implementation Status

- Integration with all Ceph daemons and clients code complete
- RBD testing possible with rados test tool (--xio)
- Finalization of open issues for full cluster testing in progress

Getting Ceph with Accelio

In official Accelio repository, use the for_next branch (includes API extensions prototyped for XioMessenger, and known to work with the published branches).

https://github.com/accelio/accelio

Branches on CohortFS Ceph repository (pre-merge) provided for Autotools and CMake build systems.

https://github.com/linuxbox2/linuxbox-ceph

- automake, xio-firefly
- cmake (what we use), xio-firefly-cmake

Accelio-specific Config Parameters

- rdma local (ip addr) : bind to specific RDMA interface
- cluster_rdma (boolean) : use XioMessenger as default cluster messenger
- client_rdma (boolean) : in clients, use XioMessenger as client messenger

example

```
rdma local = 10.17.23.10
cluster_rdma = true
client_rdma = true
```


XioMessenger Limitations

- no CephX (not yet scoped)
- flow control/recovery semantics (in progress)

XioMessenger Upstreaming Status

- in progress (wip-xio)

6/10/2014 Matt Benjamin matt@cohortfs.com

Q/A