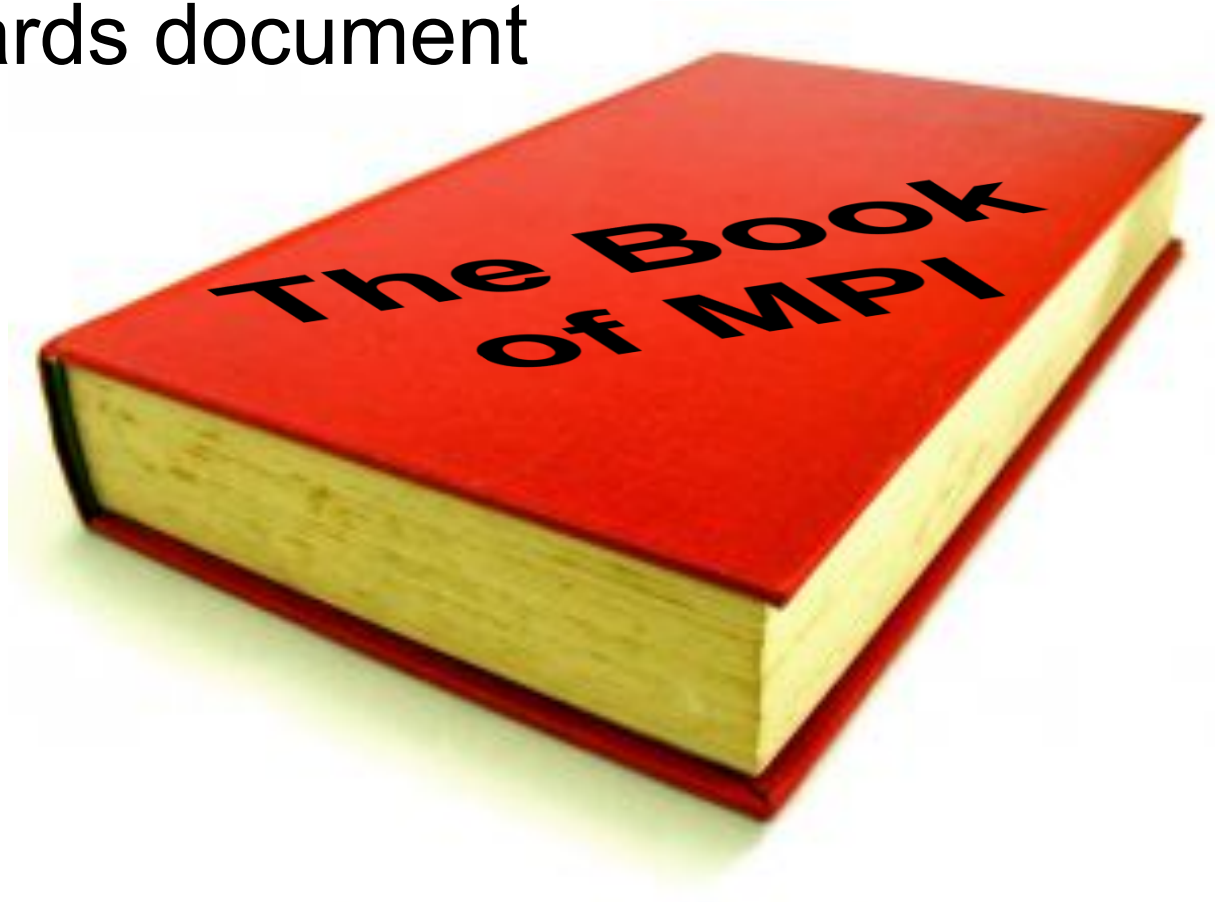
The State of libfabric in Open MPI

Jeffrey M. Squyres
jsquyres@cisco.com
16 March 2015

What is the Message Passing Interface (MPI)?

A standards document



Using MPI

Hardware and software implement the interface in the MPI standard (book)

MPI implementations

There are many implementations of the MPI standard

Some are closed source

Others are open source

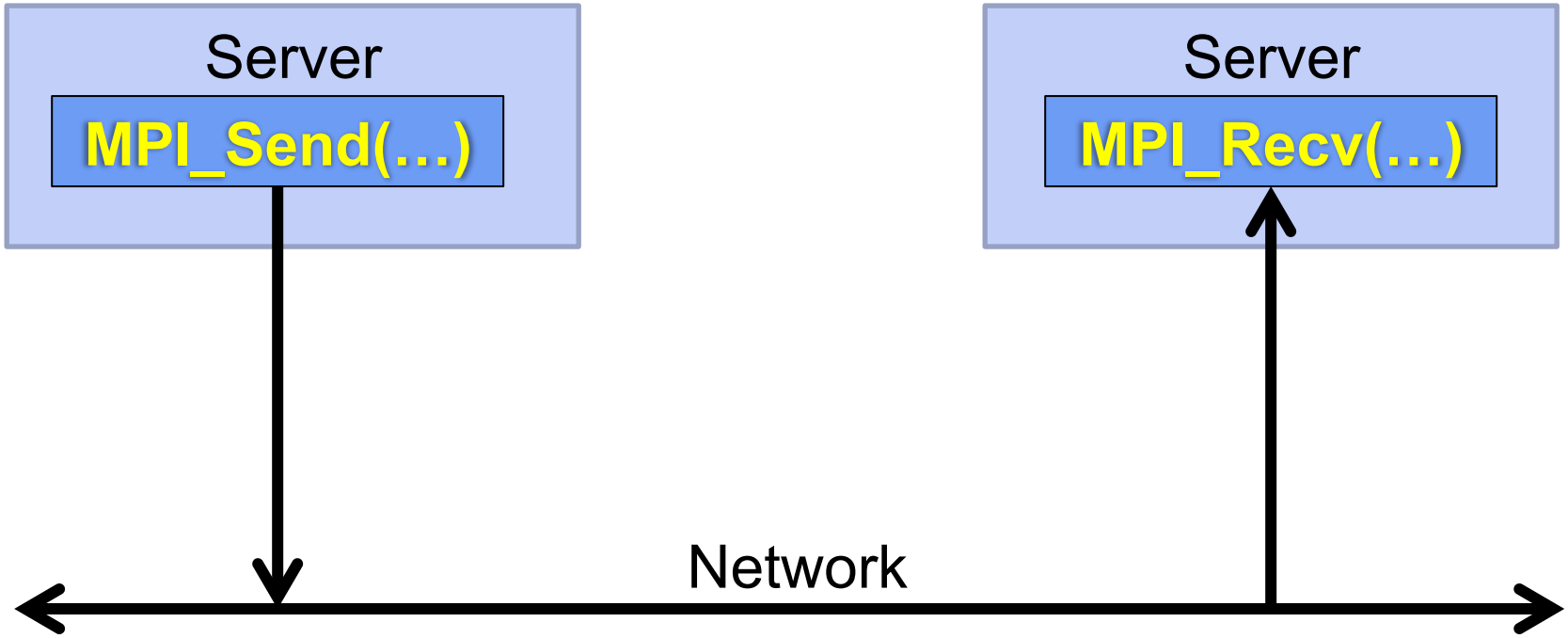
Open MPI

Open MPI is a free, open source implementation of the MPI standard

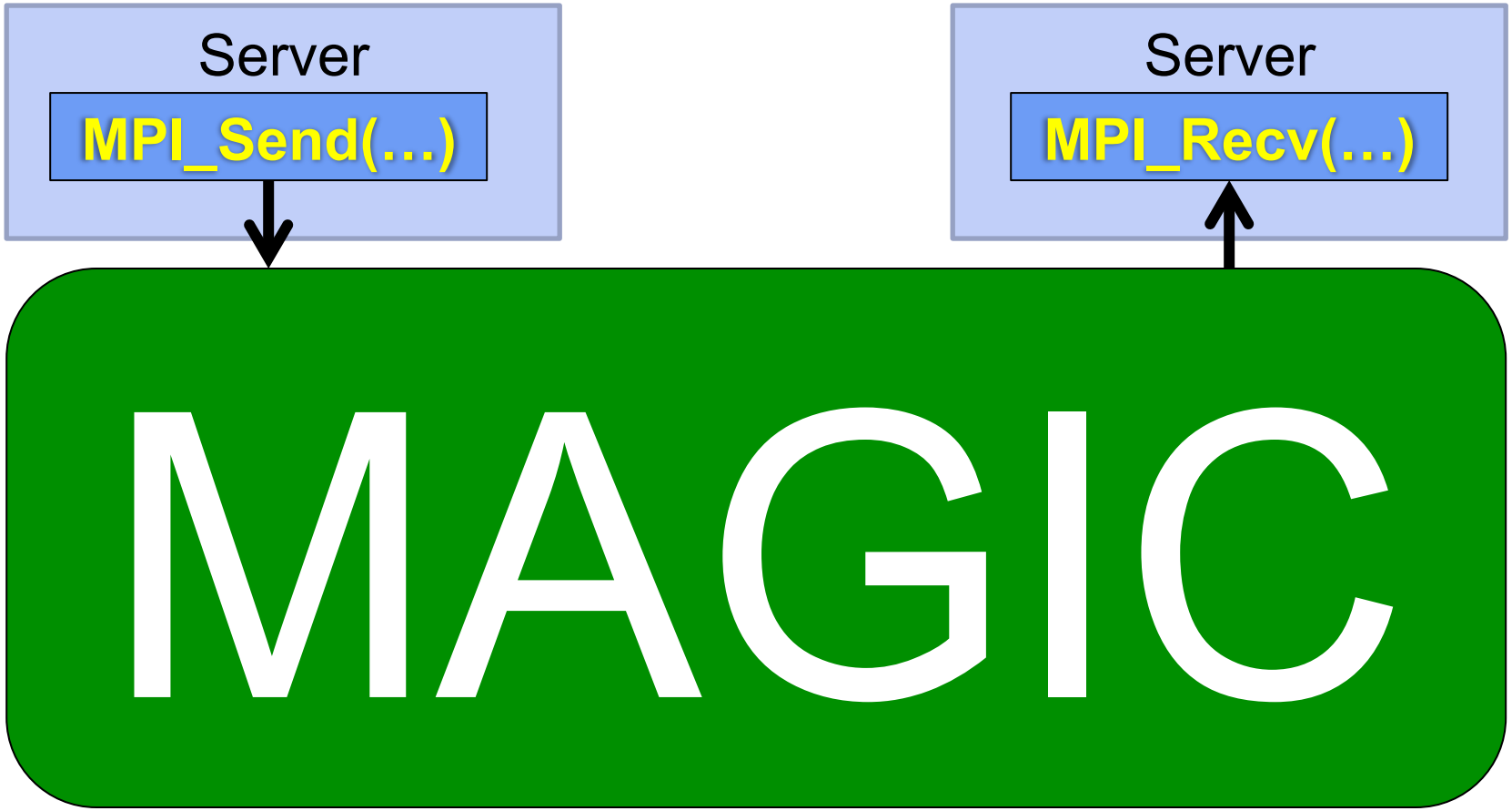
www.open-mpi.org

How I think of MPI

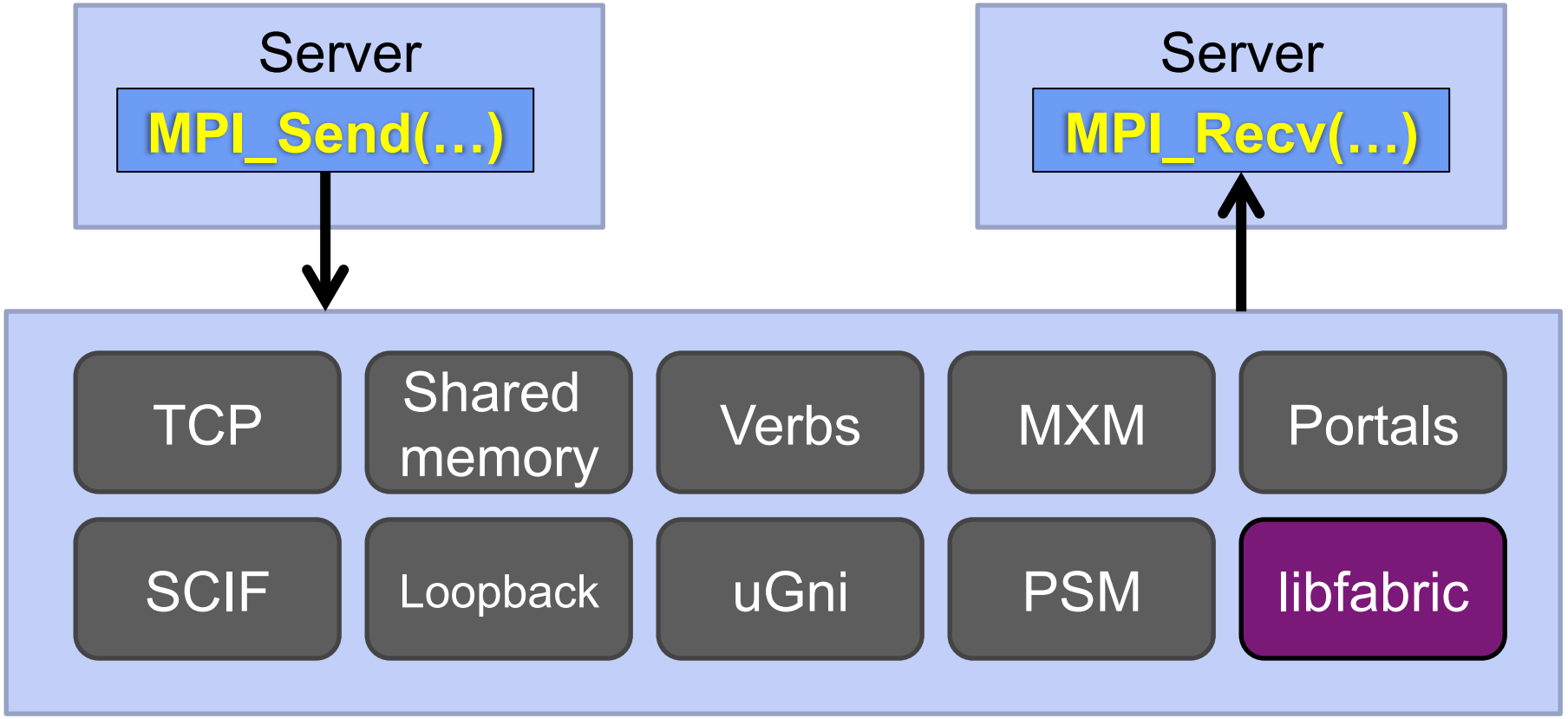
MPI abstracts away the underlying network



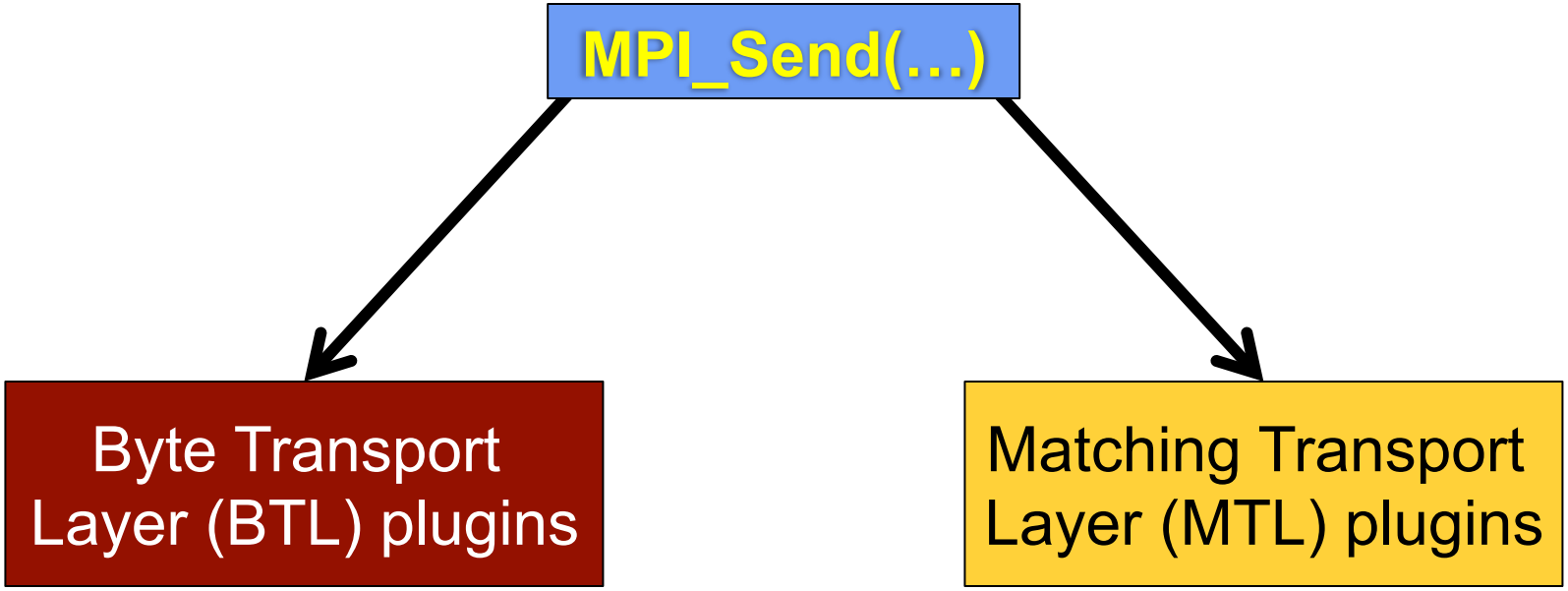
MPI abstracts away the underlying network



Open MPI multiplexes to the underlying network stack



Two major types of transports



BTL

- Inherently multi-device
  - Round-robin for small messages
  - Striping for large messages
- Major protocol decisions and MPI message matching driven by an Open MPI engine

Byte Transport Layer (BTL) plugins

MTL

- Most details hidden by network API
  - MXM
  - Portals
  - PSM
- As a side effect, must handle:
  - Process loopback
  - Server loopback (usually via shared memory)

Matching Transport Layer (MTL) plugins

BTL and MTL plugins

Byte Transport Layer (BTL) plugins

- IB / iWarp (verbs)
- Portals
- SCIF
- Shared memory
- TCP
- uGNI
- usNIC (verbs)

Matching Transport Layer (MTL) plugins

- MXM
- Portals
- PSM

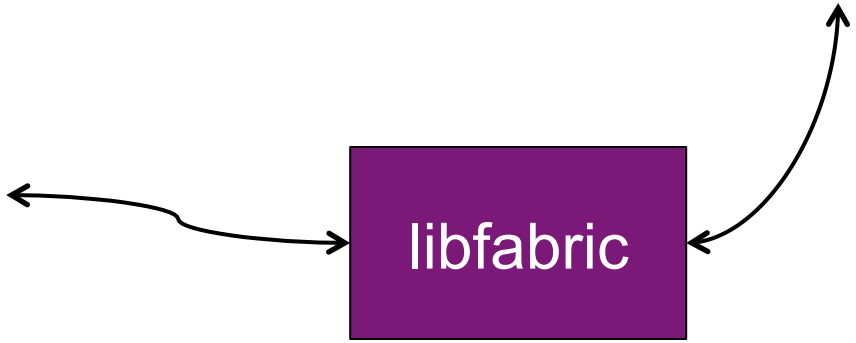
Now featuring 200% more libfabric

Byte Transport Layer (BTL) plugins

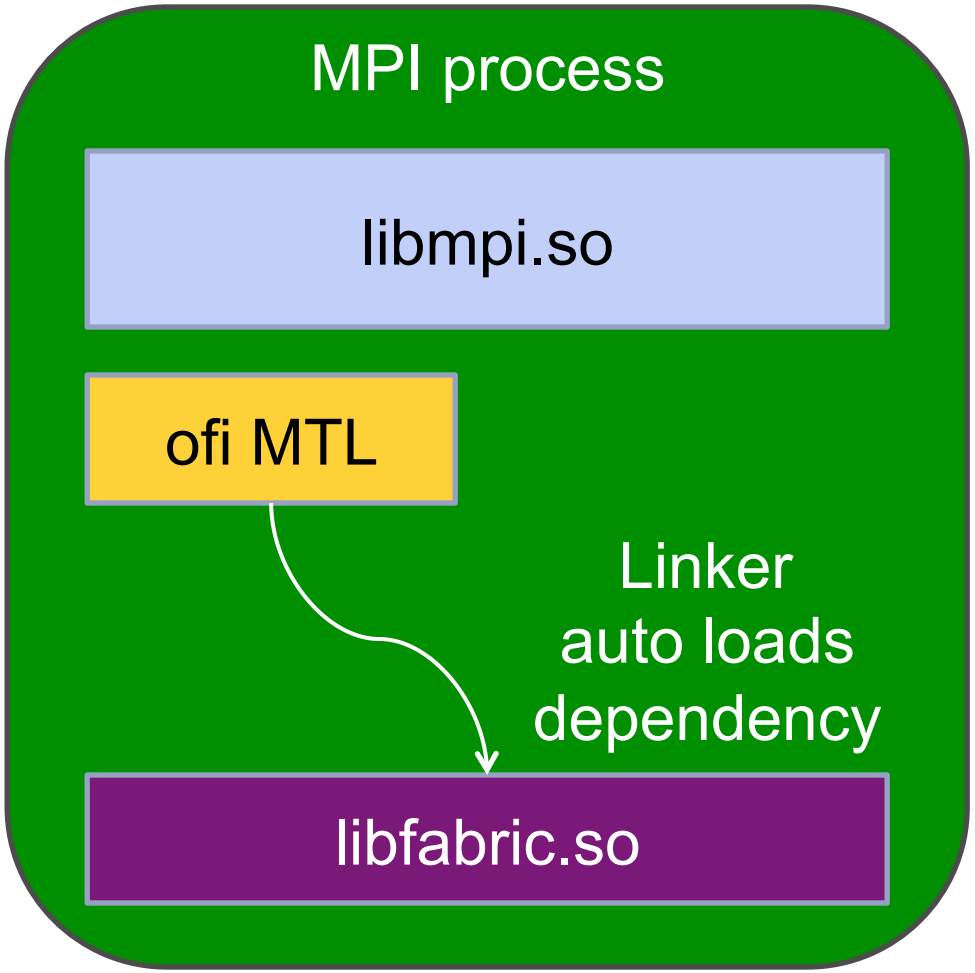
- IB / iWarp (verbs)
- Portals
- SCIF
- Shared memory
- TCP
- uGNI
- usNIC

Matching Transport Layer (MTL) plugins

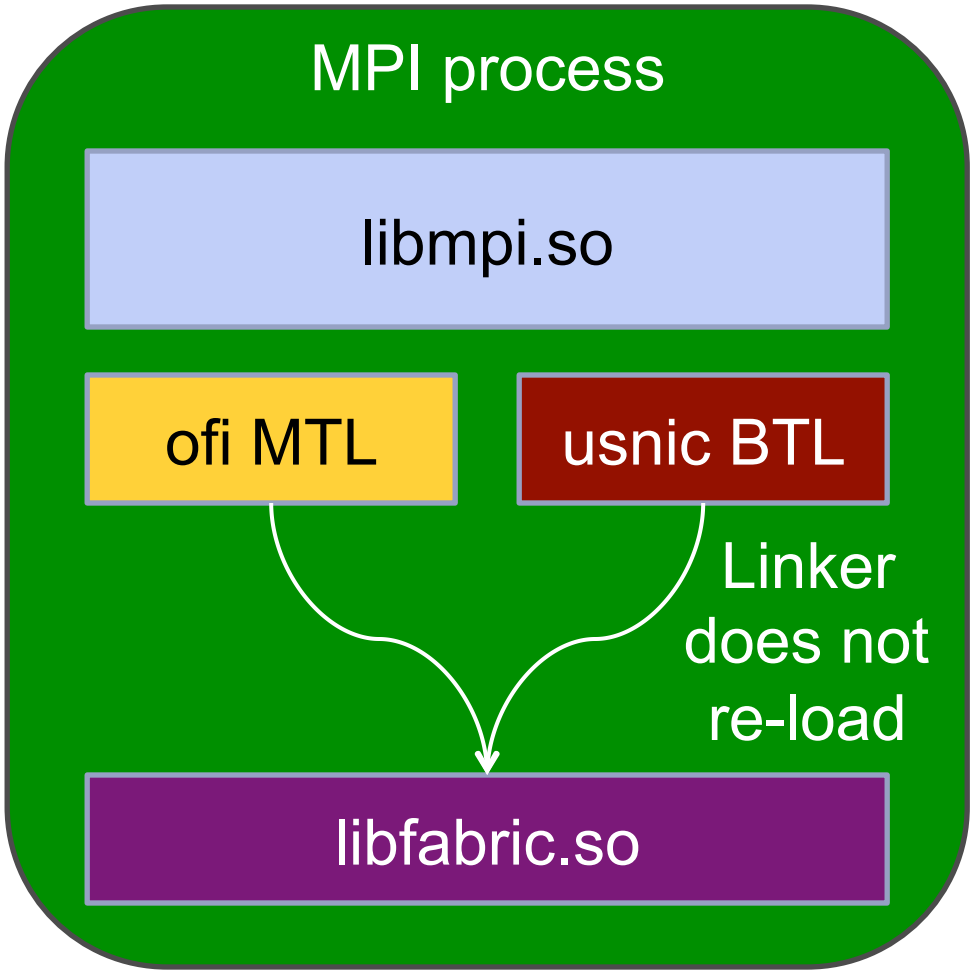
- MXM
- Portals
- PSM
- ofi



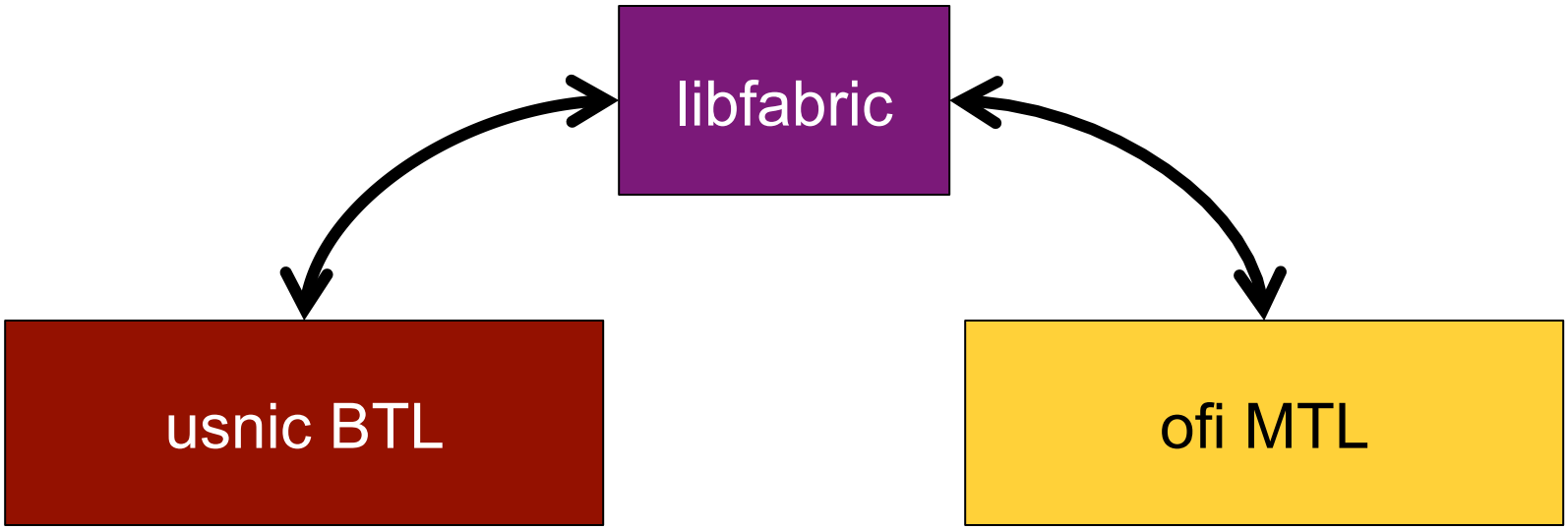
Linux linker: fun fact



Linux linker: fun fact



Libfabric-based plugins



- Cisco developed
- usNIC-specific
- OFI point-to-point / UD
- Tested with usNIC

- Intel developed
- Provider neutral
- OFI tag matching
- Tested with PSM

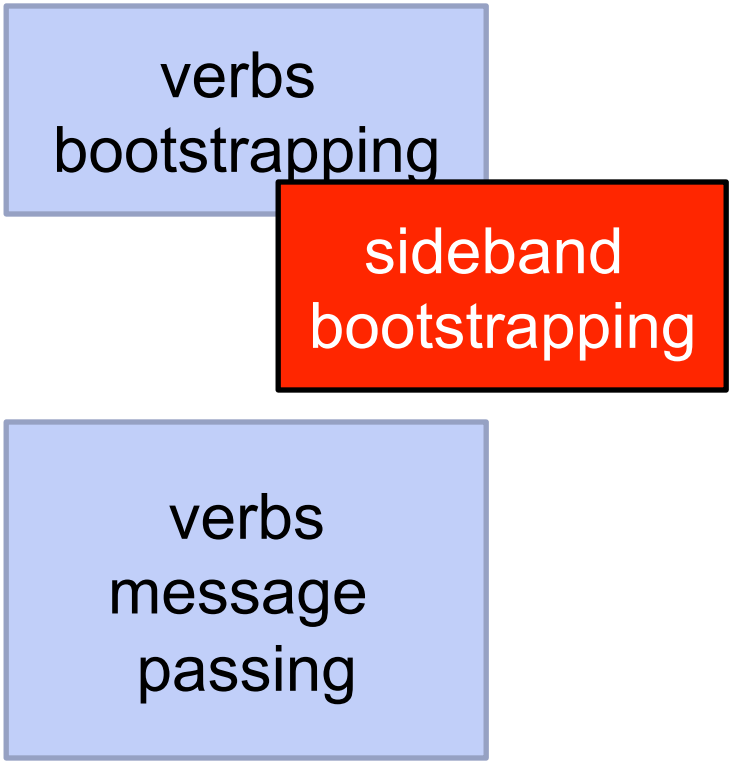
First experiment
usnic BTL: verbs → libfabric

verbs bootstrapping

verbs message passing

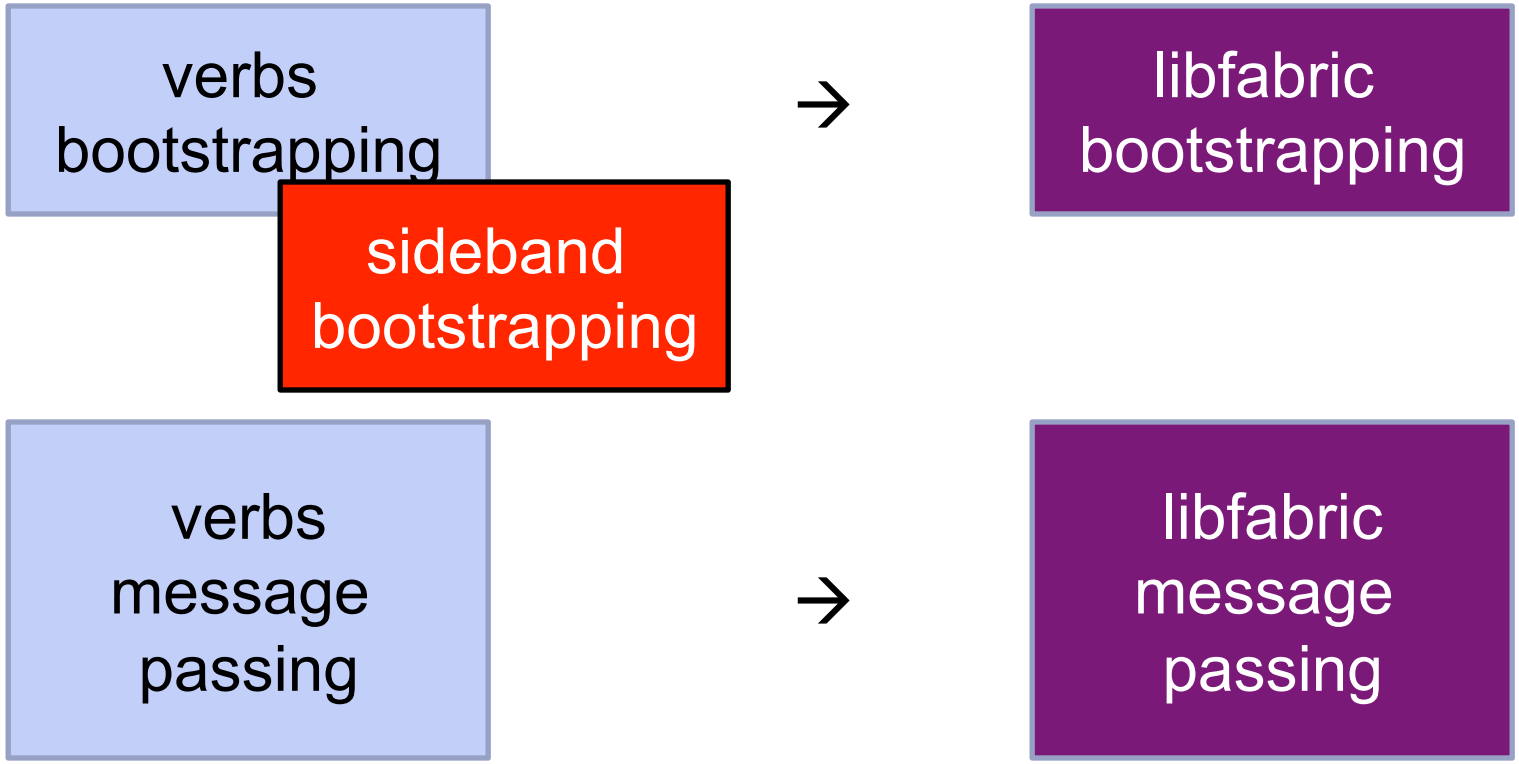
Can loosely classify the usnic BTL into two parts

First experiment
usnic BTL: verbs → libfabric



1. Find the corresponding ethX device
2. Obtain MTU
3. Open usNIC-specific configuration options

First experiment
usnic BTL: verbs → libfabric



Comparison results

Bootstrapping sequence totally different

libfabric requires no sideband bootstrapping

Pretty much a 1:1 swap
of verbs → libfabric calls

Second experiment
Two different libfabric usage models

usnic BTL

- For a specific provider
  - Ask fi_getinfo() for prov_name="usnic"
- Use usNIC extensions
  - Netmask, link speed, IP device name, etc.
- usNIC-specific error messages

ofi MTL

- For any tag-matching provider
- No extension use
  - 100% portable
- Generic error messages

Second experiment
Two different libfabric usag[illegible]odels

usnic BTL

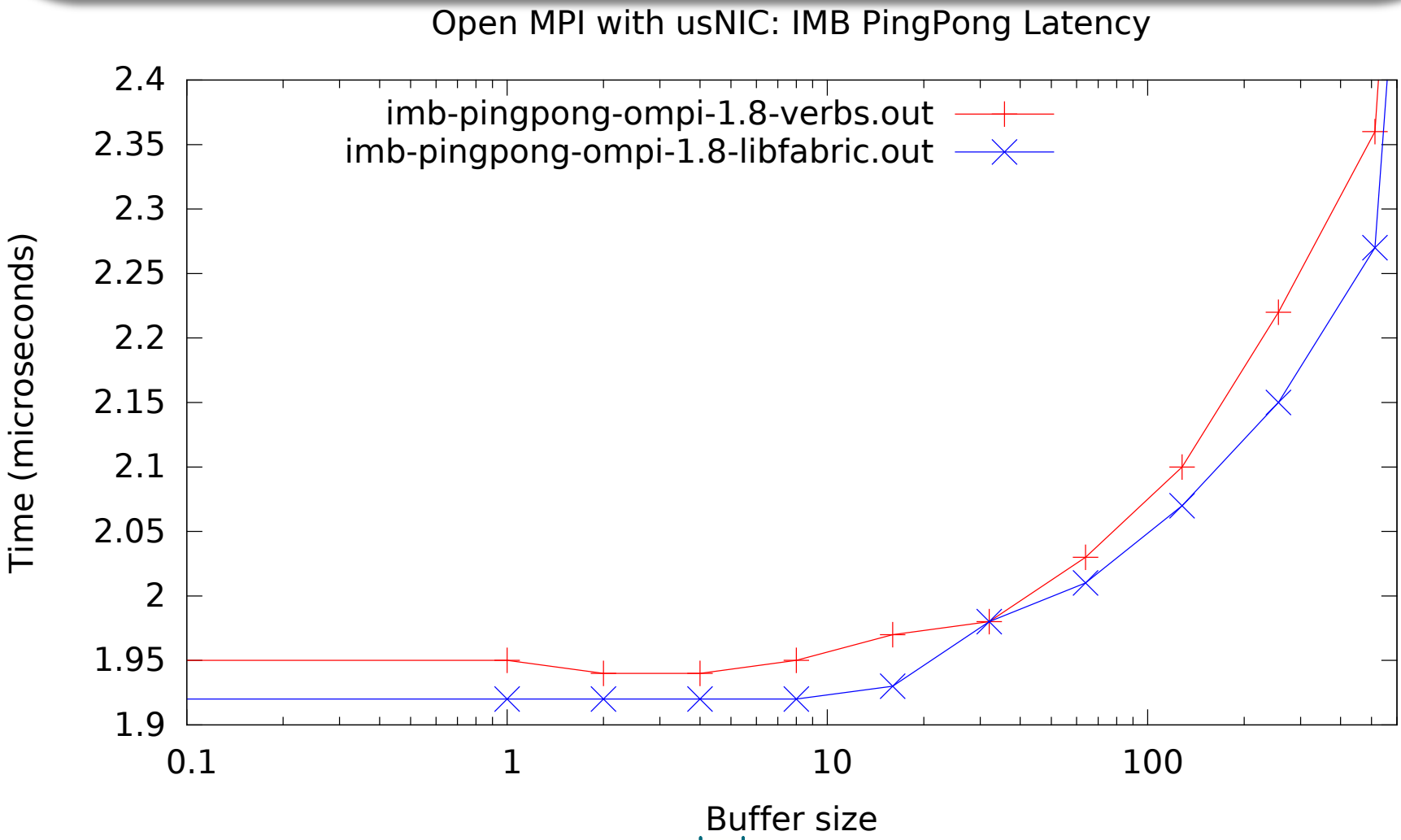
[illegible]

- For a s[illegible]rovider
  - Ask[illegible]) for prov[illegible]snic"
- Use u[illegible]
  - Netr[illegible] IP de[illegible]
- usNIC-s[illegible]ror messages

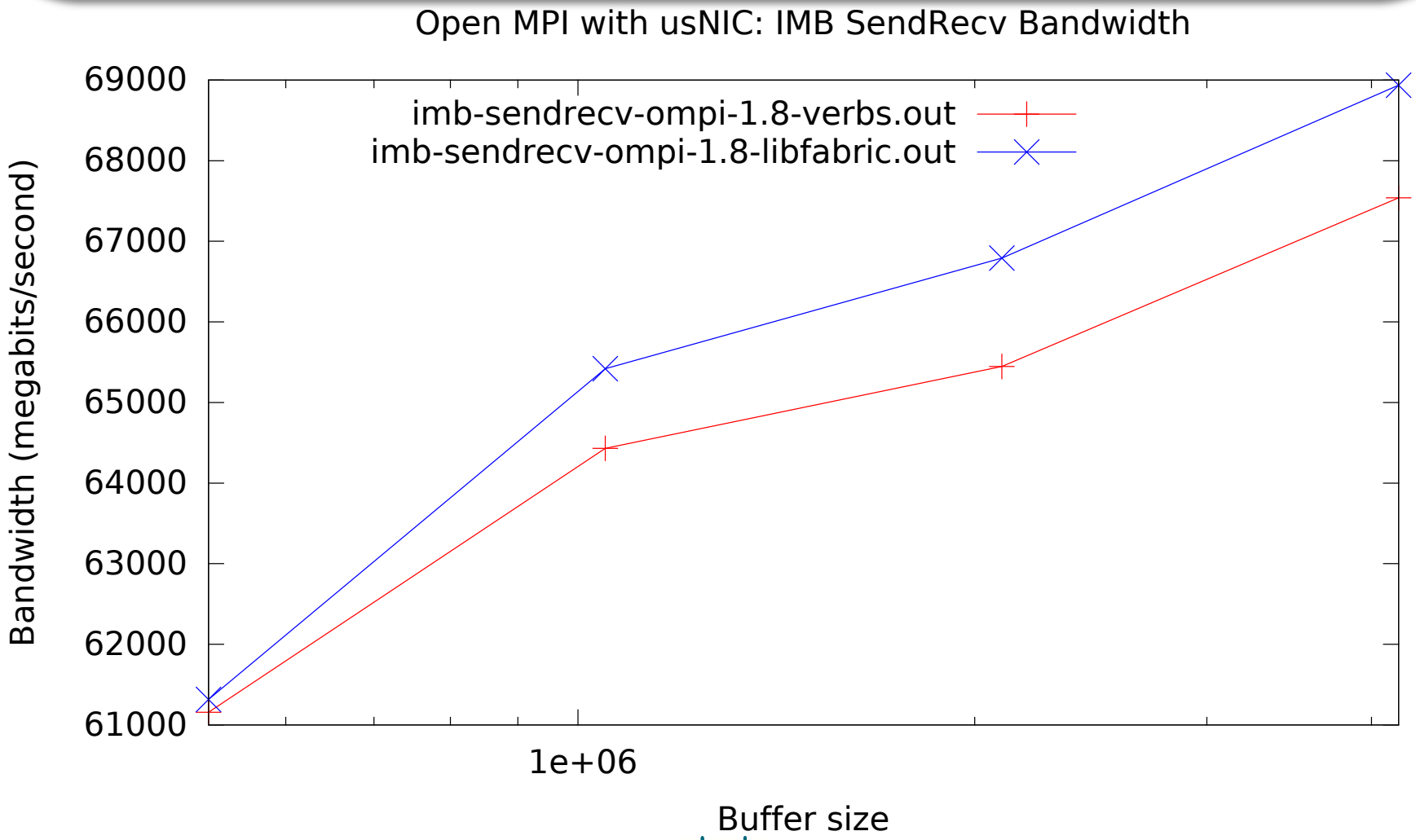
[illegible] tag-matching
[illegible]der
[illegible]o extension use
  - 100% portable
- Generic error messages

100% libfabric goodness

libfabric performance vs. Linux verbs

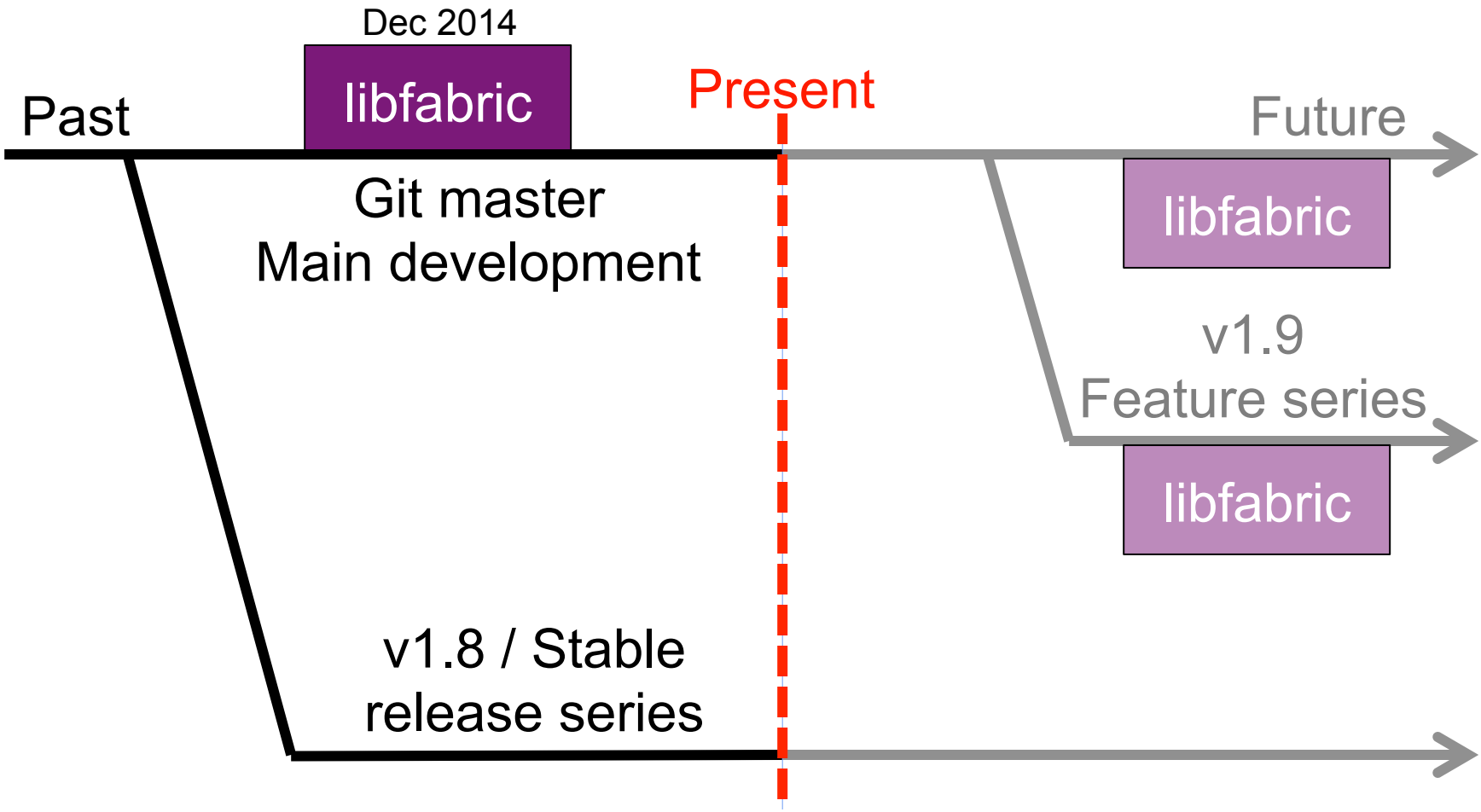


libfabric performance vs. Linux verbs

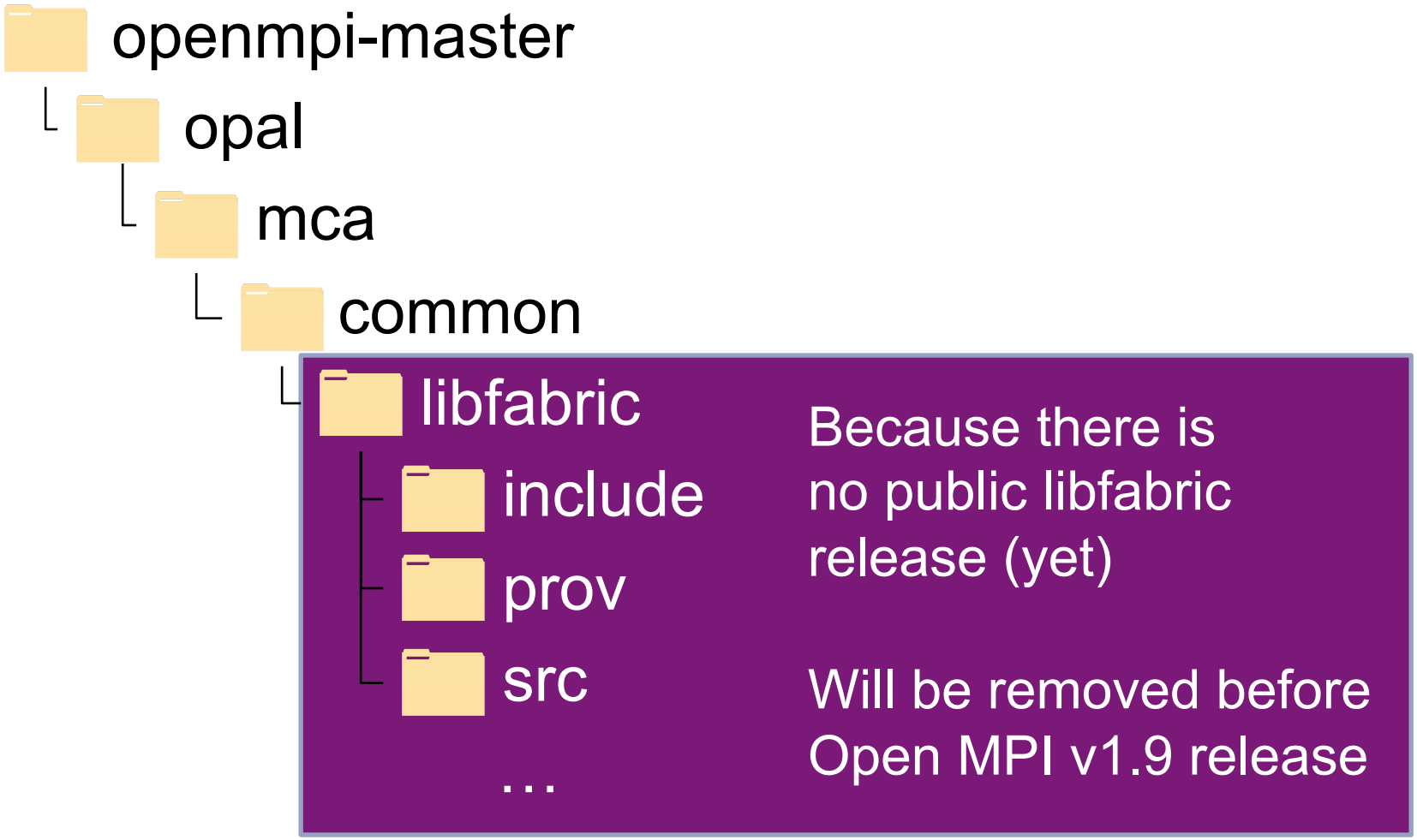


Version roadmap

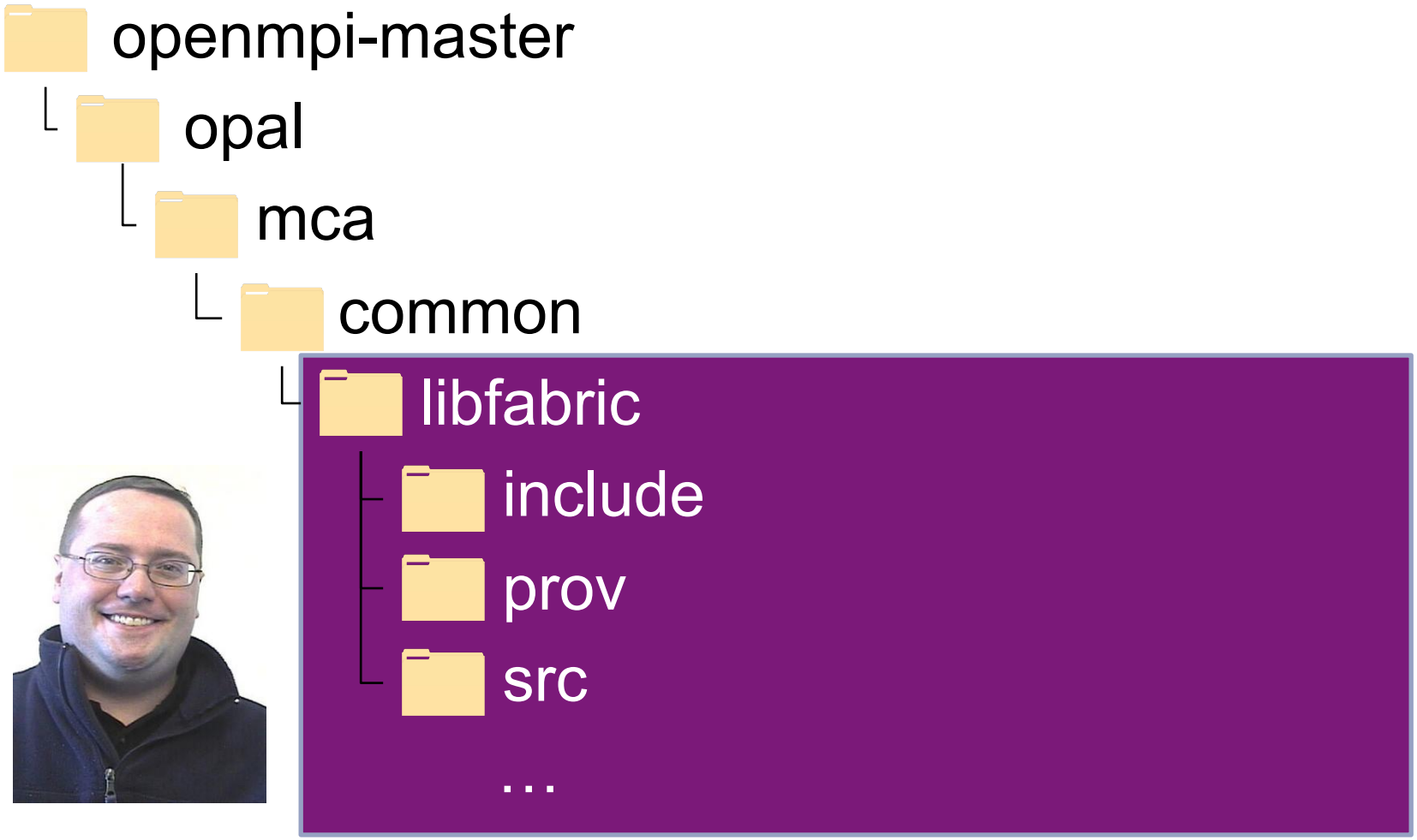
Version roadmap



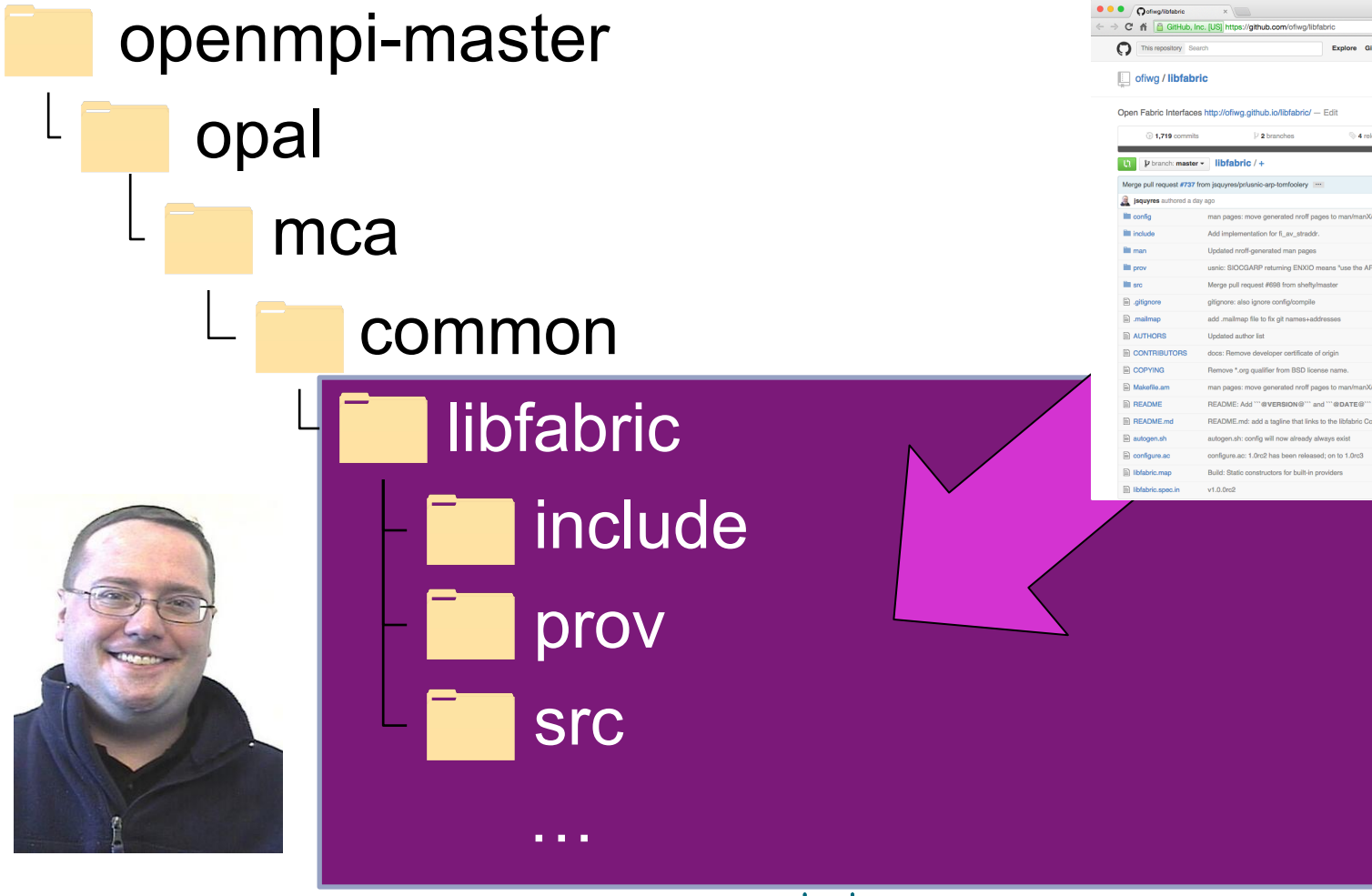
Currently embedding libfabric

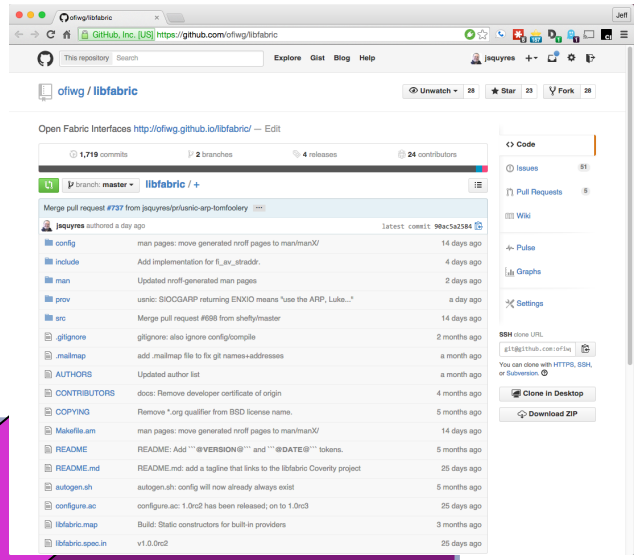


Periodic refresh from libfabric Github



Periodic refresh from libfabric Github



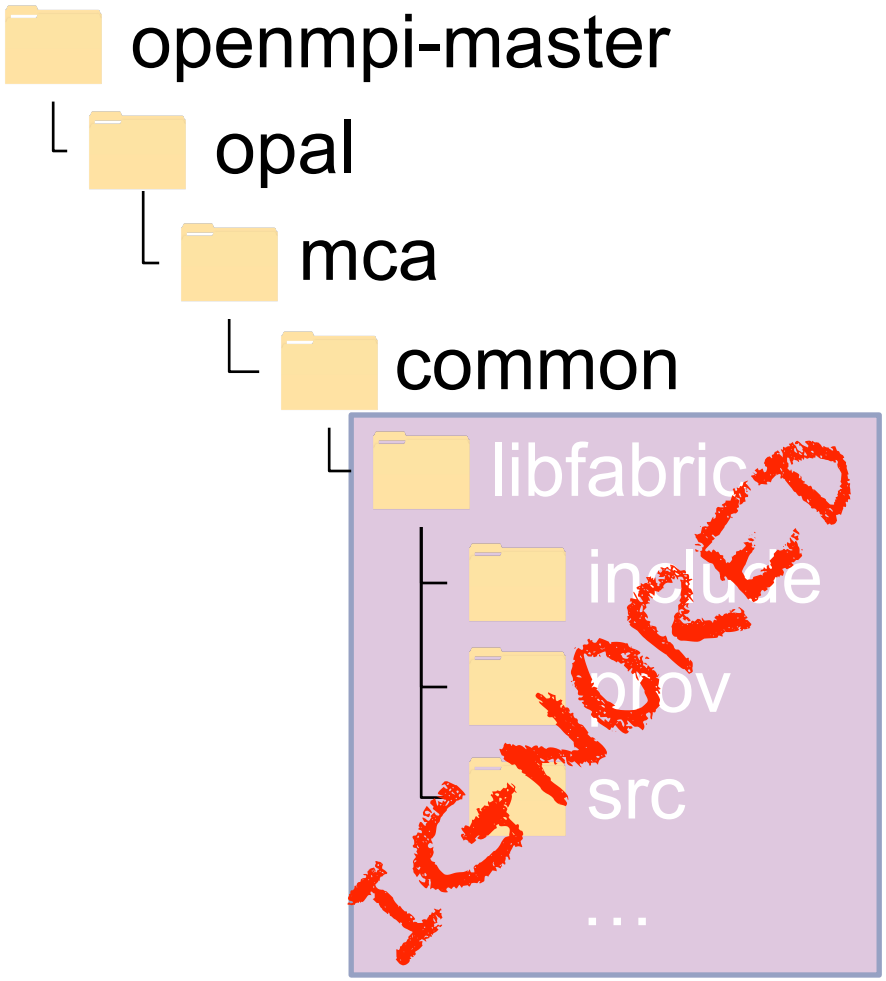


Periodic refresh from libfabric Github

- openmpi-master
  - opal
    - mca
      - common
        - libfabric
          - include
          - prov
          - src
          - …

Moar new libfabric goodness!

Can also build against external libfabric



libfabric

(e.g., installed under $HOME, or in /usr, or …)

Will be the only model in v1.9

openmpi-master

libfabric

(e.g., installed under $HOME, or in /usr, or …)

Feedback loop = good

- Using libfabric in its (first) intended environment was quite useful
  - Resulted in libfabric pull requests, minor changes, etc.
- Biggest thing missing is the mmunotify functionality
  - …will file a PR/RFC about this soon

Questions?