Photon: Shining a light on the I/O problem in HPC

Martin Swany

University of Delaware

The Problem

- Growing I/O gap in HPC
  - Problems effectively using the network in Clusters, Clouds and Grids
- On the one hand -- Extreme scale systems
- On the other -- 100Gb/s networks
- Components scaling up but the gap is growing
  - Highlighted in various places including yesterday’s panel

One scenario

- MPI-IO to a parallel filesystem
- GridFTP servers mount this filesystem and perform parallel file transfers
- Data has been forced into a sequential file
- Did the parallelism in the program match that of the object stores in the filesystem? Does the GridFTP striping match it?
- “Optimized” separately (if at all)
    - Even separately, the optimizations are inadequate

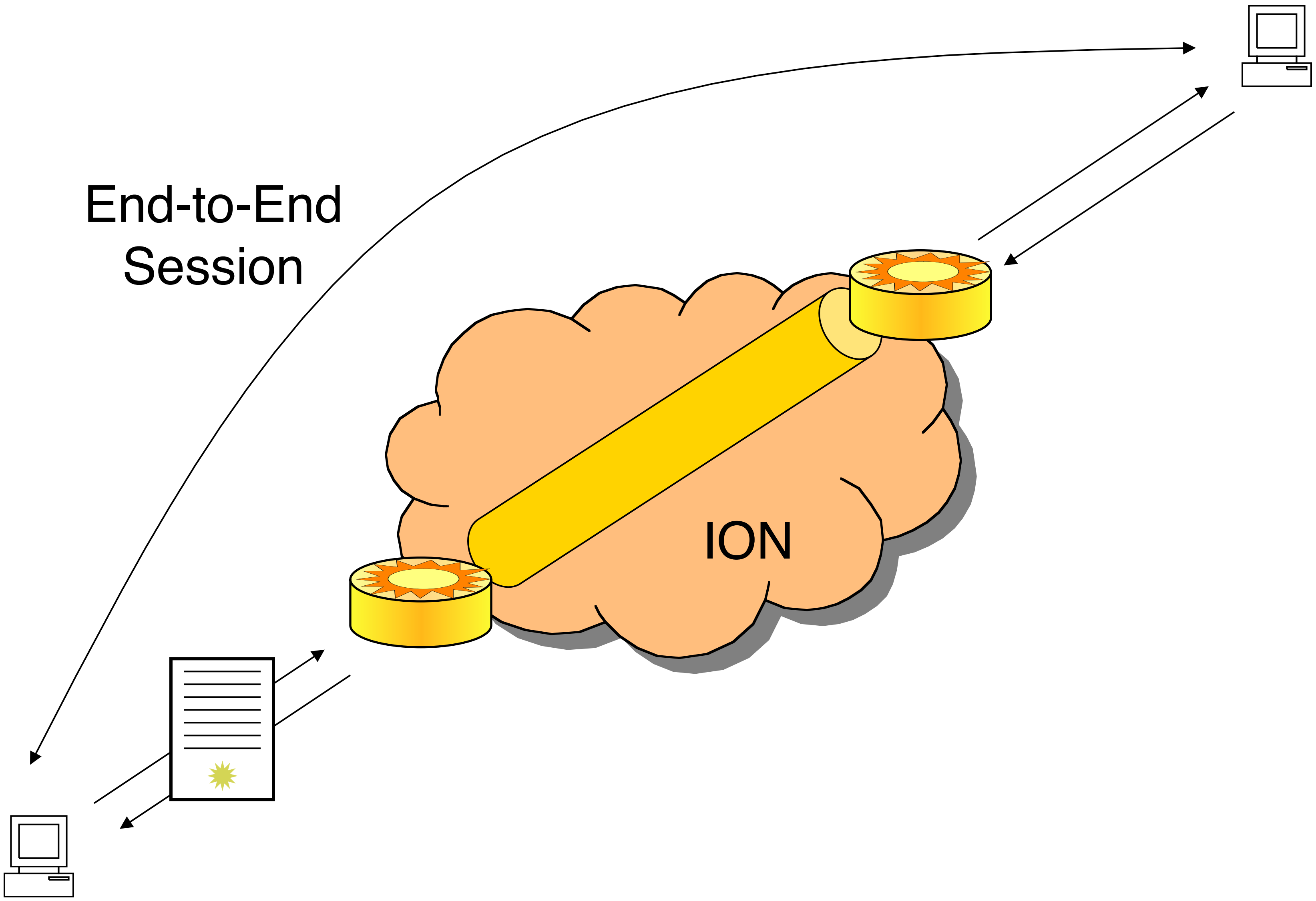
Photon

- Unifies previous solutions into an end to end system for parallel I/O
- Wide-area data movement
  - Phoebus
- MPI program transformation
  - AToMS
- Lightweight cluster data movement
  - Gravel -> Photon
- Deconstructed filesystems
  - eXnode

Phoebus

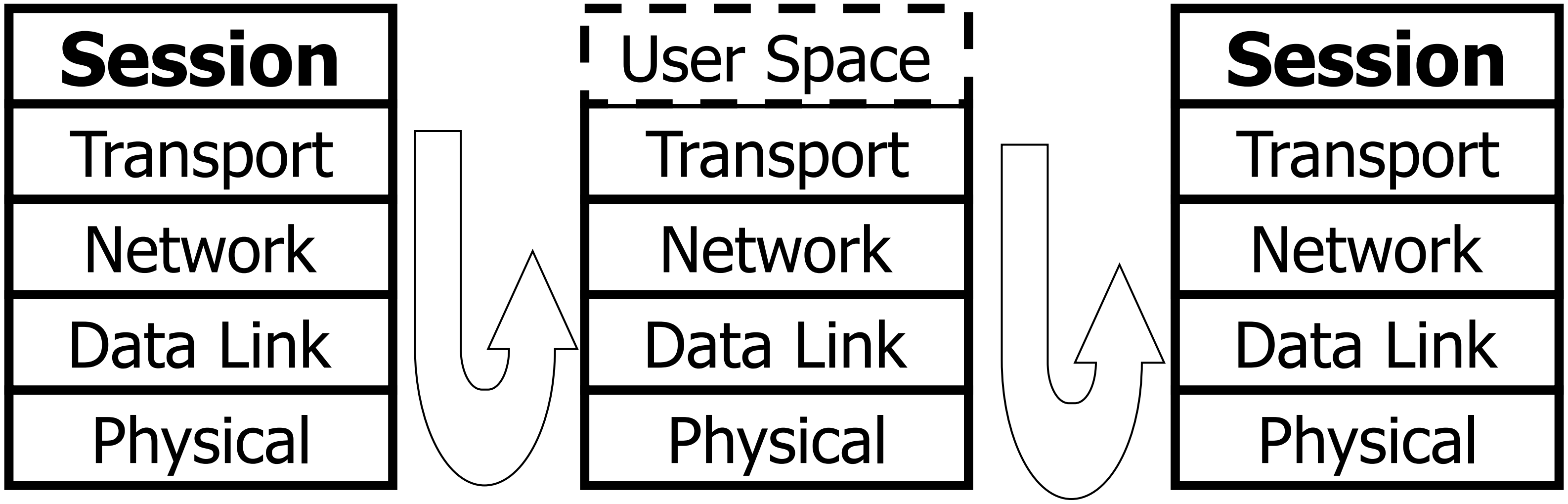
- The Phoebus project aims to bridge the network performance gap by providing an optimizing network service
- Phoebus is based on the concept of a "session" that enables multiple adaptation points in the network to be composed
- Phoebus provides a gateway for legacy applications to use advanced networks
  - Network reservation like ESnet's OSCARS, Internet2 ION

Phoebus



Session Layer

- A *session* is the end-to-end composition of *segment-specific* transports and signaling
  - More responsive control loop via reduction of signaling latency
  - Adapt to local conditions with greater specificity
  - Buffering in the network means retransmissions need not come from the source



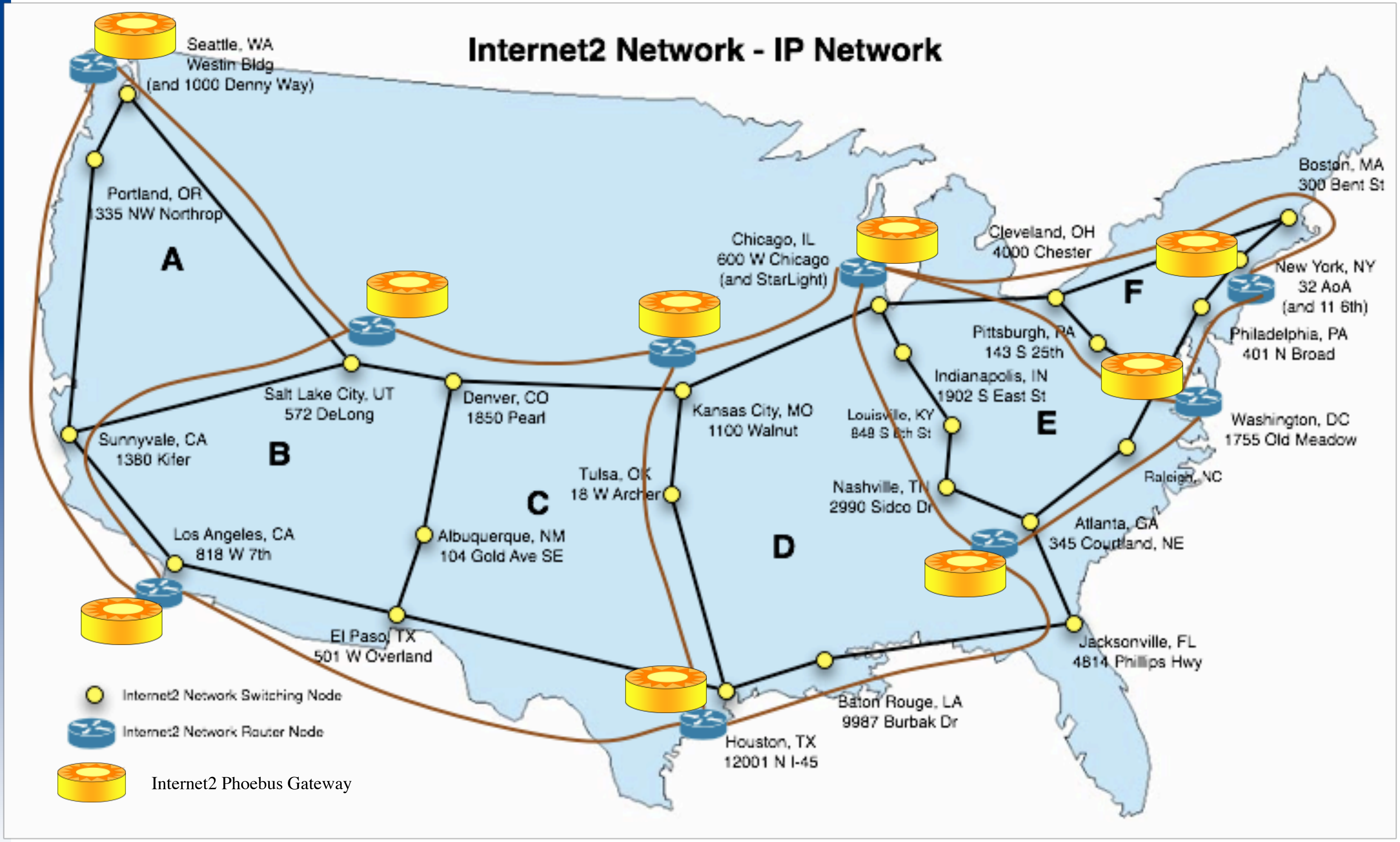
Dynamic Networks

- The last piece of the Cloud puzzle
  - Network allocation is the elephant in the cloud
- Phoebus signals dynamic networks like ESnet, Internet2, GEANT...
  - Phoebus speaks to the control plane to provision network resources
- Once the connection is established to the Phoebus node, traffic can begin to flow
  - Could be sent over an existing link if unable to provision
  - Phoebus can finish the connection over the commodity network if the allocation times out

Session Layer Benefits

- Our session-layer approach is an architectural evolution for the Internet
  - ongoing work in GENI, DOE Networks program
- A session layer provides explicit control over ***adaptation points*** in the network
  - Transport protocol
    - Rate-based to congestion based
    - Shorter feedback loops
  - Traffic engineering
    - Map between provider-specific DiffServ Code Points / VLANs
  - Authorization and Authentication
    - Rich expression of policy via e.g. the Security Assertion Markup Language (SAML)

Deployment Plans



Phoebus and GridFTP

- Integrated with GridFTP
  - *globus-gridftp-server* loads the Phoebus XIO driver when requested
  - *globus-url-copy* extended to support Phoebus-based transfers with -ph flag or explicitly with –dcstack
  - Support for advanced features
    - 3rd party transfers
    - Parallel streams
- SC09 paper and JPDC article in press show performance over 10G

ATOMS

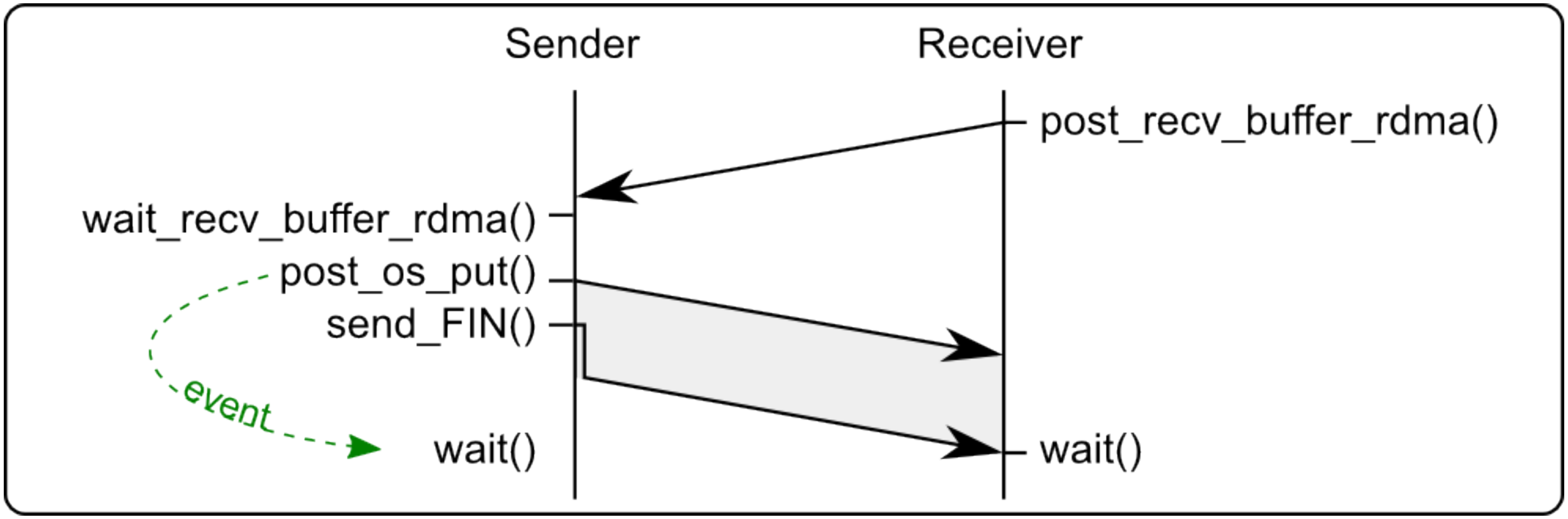
- Focused on message-passing performance in clusters
- AToMS = Auto-Tuning of MPI Software
    - Actually, Transformation and Tuning
- Kennedy’s telescoping languages work – “improves performance by replacing sequences of library calls with equivalent, but more efficient, sequences.”
- Partial implementations in Open64 and LLVM

AToMS Transformations

- Transform MPI communication
  - Collectives → Point-to-point
  - Blocking → Non-blocking
  - Non-blocking → One-sided
  - Send fission and fusion
  - Restructure (user-defined) MPI datatypes
- Separate components of communication, code motion to improve overlap
  - Memory registration
  - Metadata exchange
  - Data movement
  - Progress/Completion

Gravel

- Library for use by AToMS transformations
- RDMA put/get for data movement
  - Also MX
- “Ledger” for progress and synchronization
  - Also available via RDMA

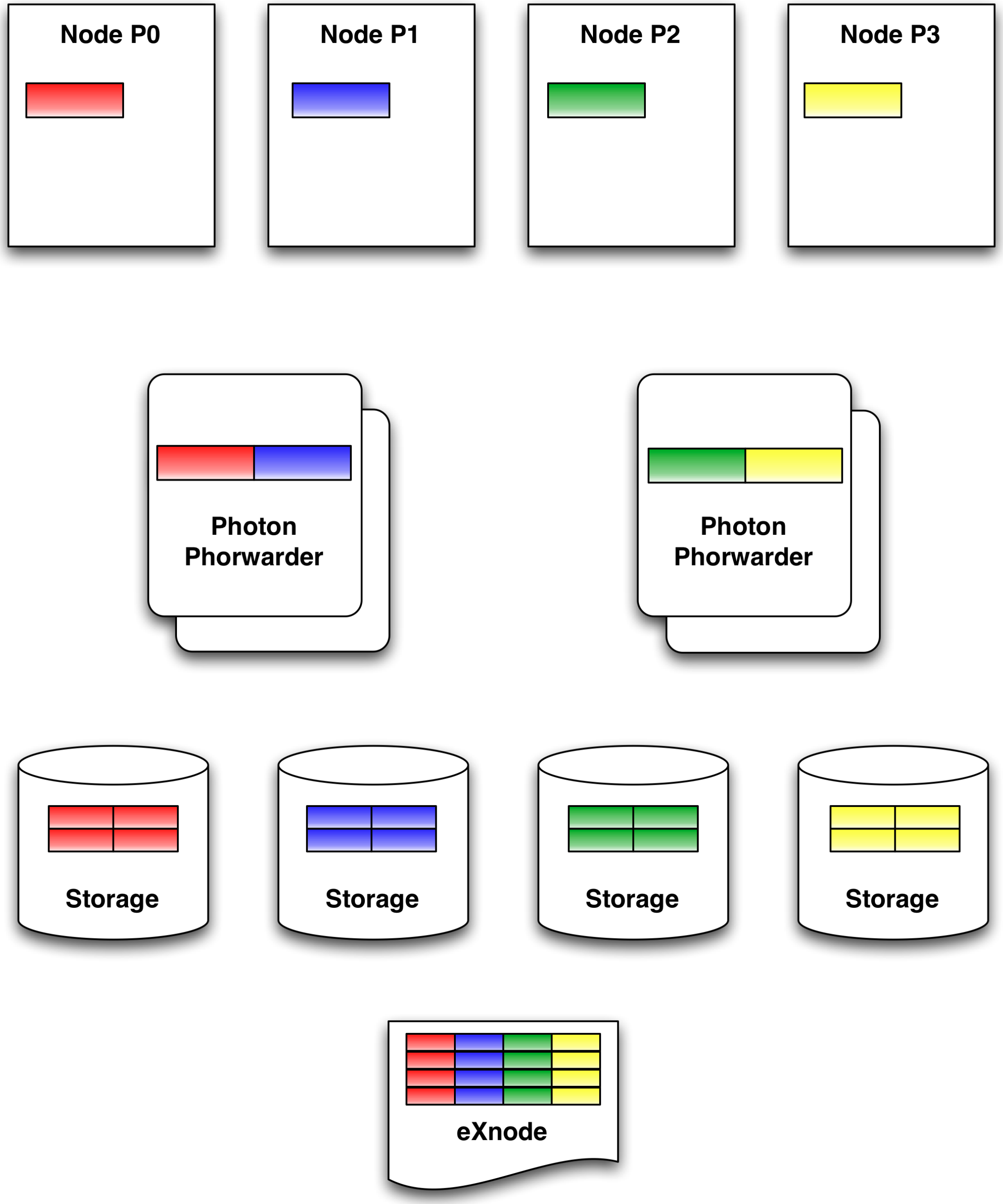


eXnode

- Concept from Logistical Networking work (IBP)
- Analogous to a filesystem inode, but available in the application
- Describes location and relationship of blocks in a (virtual) file
- Allows us to provide filesystem-like semantics without filesystem overhead
- Data “chunks” in the eXnode can refer to IBP allocations, Photon buffers, etc

Photon

- Transform MPI-IO calls into Photon calls
  - pattern, location are often discoverable and consistent over the lifetime of the application
- Create data movement session with various intermediate forwarders
  - Similar ideas to those in DART, ZOID
- This takes advantage of our session protocol and asynchronous progress notification
  - “For the next 100K iterations, watch the ledger for completion, grab the data, update the completion ledger”
- Unify wide-area and cluster-area optimizations
  - Building on the mature Phoebus forwarder


Node P0
Node P1
Node P2
Node P3
Photon Phorwarder
Photon Phorwarder
Storage
Storage
Storage
Storage
eXnode

Implications for next-generation systems

- Programs need APIs, with flexibility of implementation
- In particular, when library or OS functionality can be inlined in the application, we expose more opportunities to mitigate latency
- Don't force a given application to pay for what it doesn't need – "deconstruction"
  - E.g., POSIX I/O with filesystem semantics
- Single framework enables optimizations

End

- Thank you for your attention
- Questions?