# EXASCALE … AND BEYOND

DEHPC 2015

David Lombard

Software Architect Extreme Scale Computing, Senior Principal Engineer, Intel Corporation

October 18, 2015



THE NEW CENTER OF POSSIBILITY

# LEGAL DISCLAIMER

Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel, Intel Xeon, Intel Core microarchitecture, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others

# KEY CHALLENGES TO EXASCALE

Power
- Many aspects drive and are driven by power

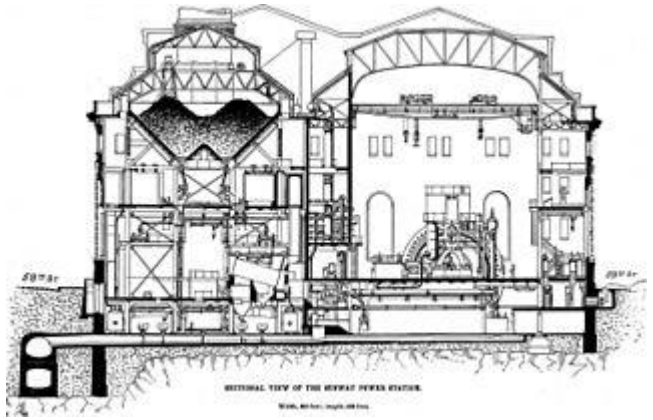
Scalability
- Forcing hybrid programming models and hierarchy system software

Reliability
- Application and system software not designed to handle faults

Communication bandwidth and latency
- Induces challenges in existing BSP model

Memory bandwidth, latency, and capacity
- Pushes on threading and increased need for parallelism

# POWER



## An increasing concern

- TN8: power band and ramp
  - Lower-bound a whole site issue
- HPC Power API Spec
  http://powerapi.sandia.gov/

## Energy efficiency

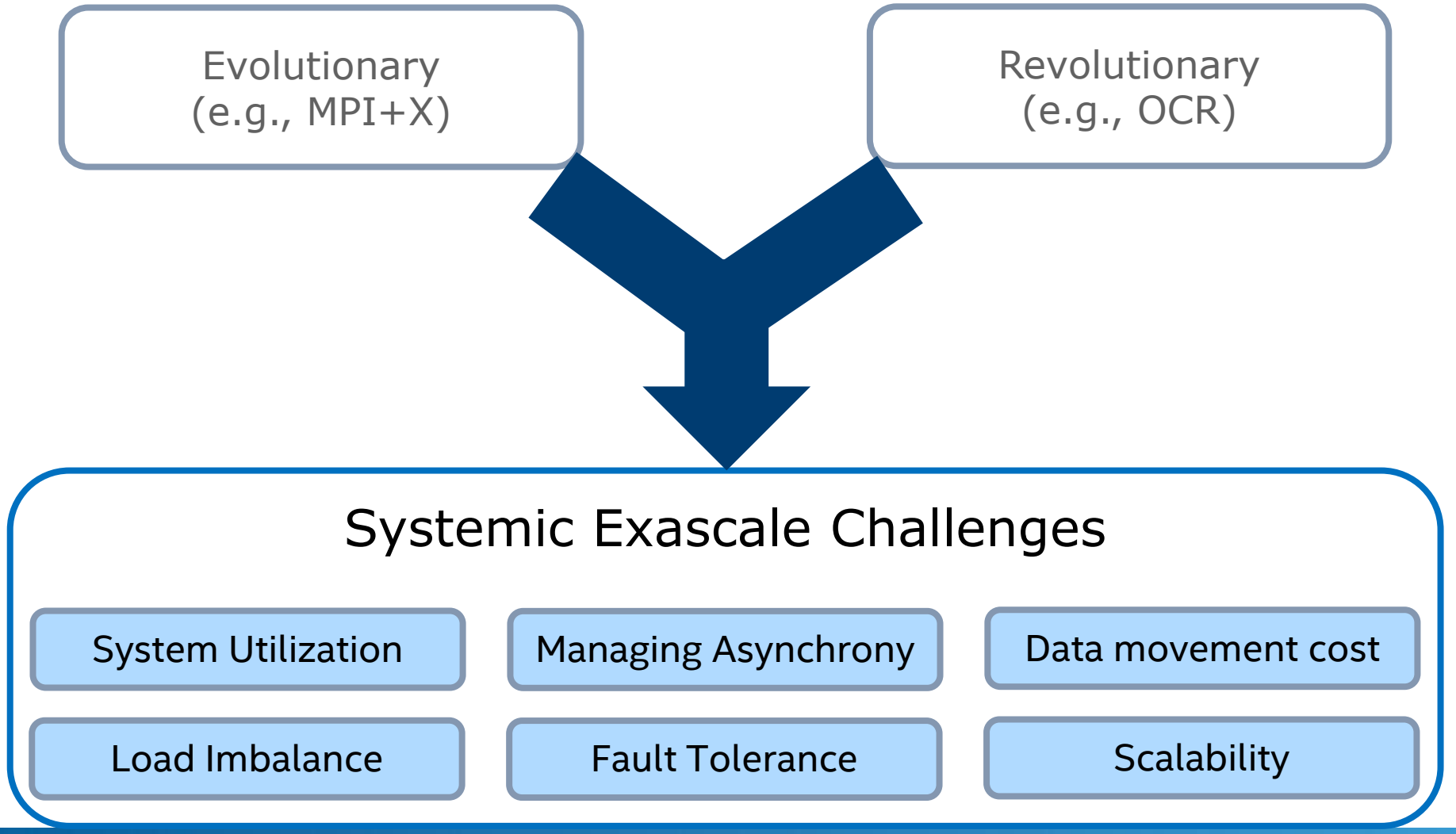
- A prime focus at LRZ
- Will become more common?

## Tools

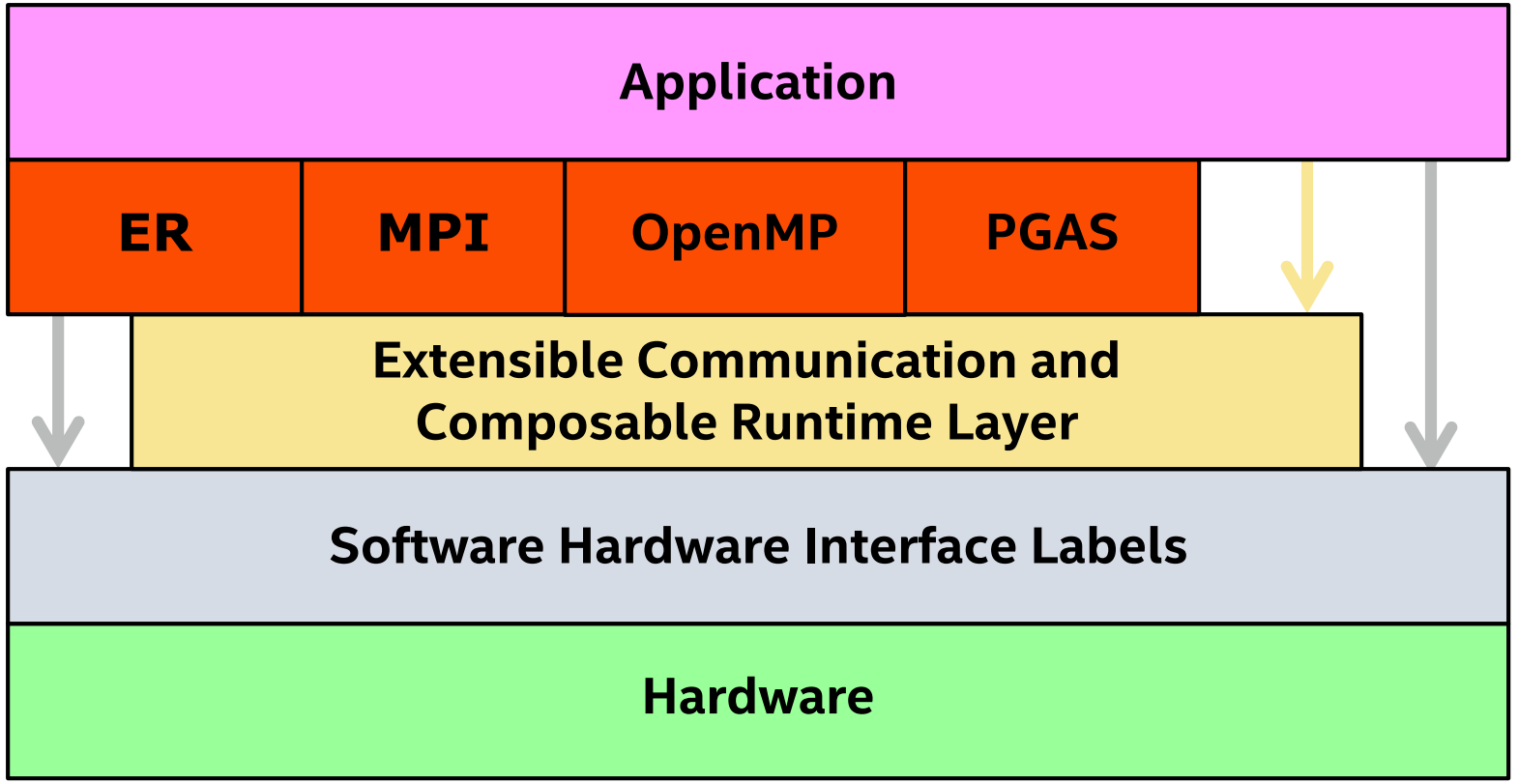
- Energy efficiency for clients/servers
  - Sleep as much as possible
    → PowerTop
    https://01.org/powertop
- Energy efficiency for HPC
  - Understanding the code
    - Global observation v. tuning
  - Manage performance to achieve goals
    → Global Energy Optimization
    https://geopm.github.io/geopm/

    Just posted; feedback on interfaces, docs

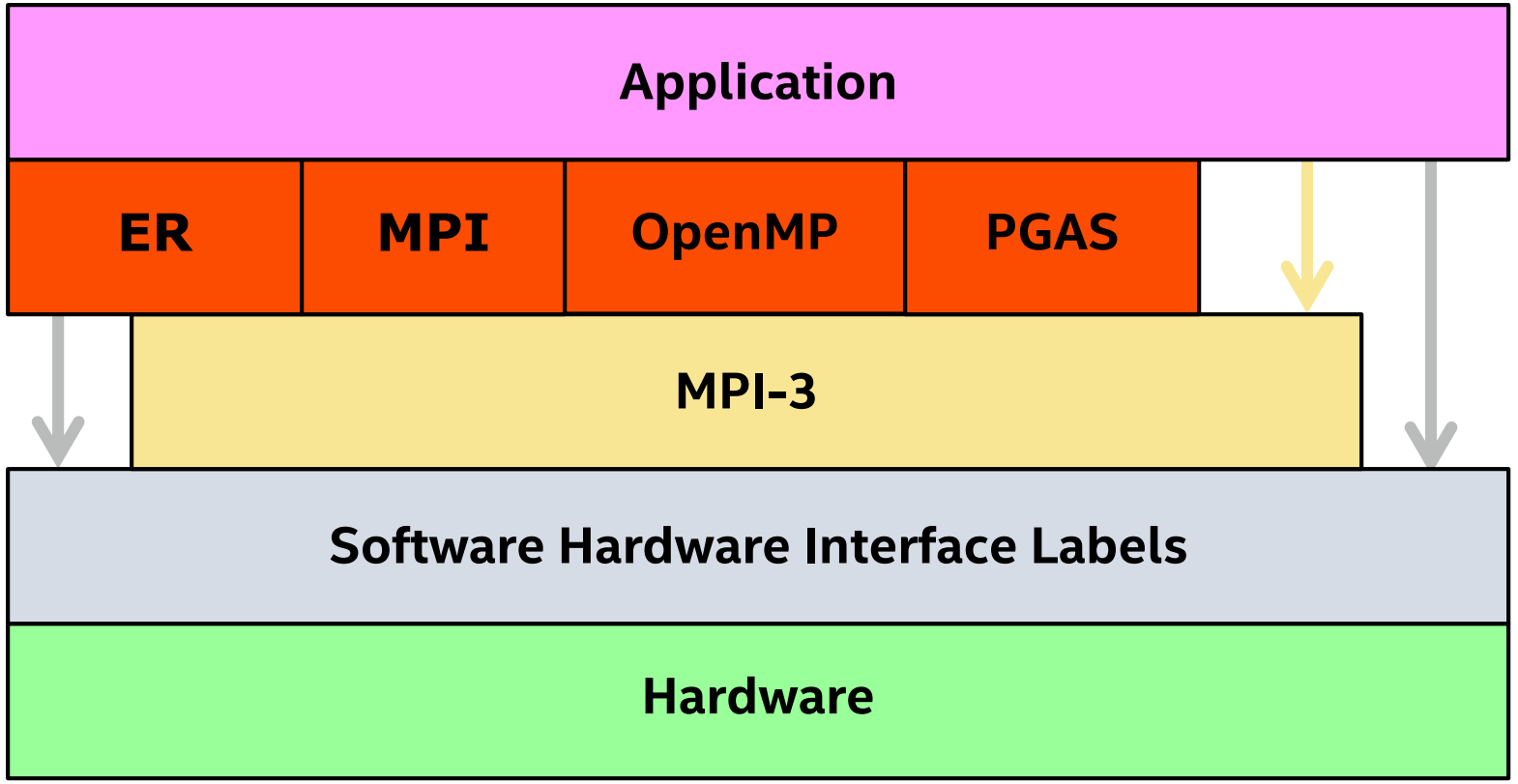
# PATHWAYS TO EXASCALE



# PROGRAMMING MODEL COMPOSABILITY



# PROGRAMMING MODEL COMPOSABILITY




Non-blocking collectives, topology, one-sided comms, POSIX/Sys5 SHMEM, …

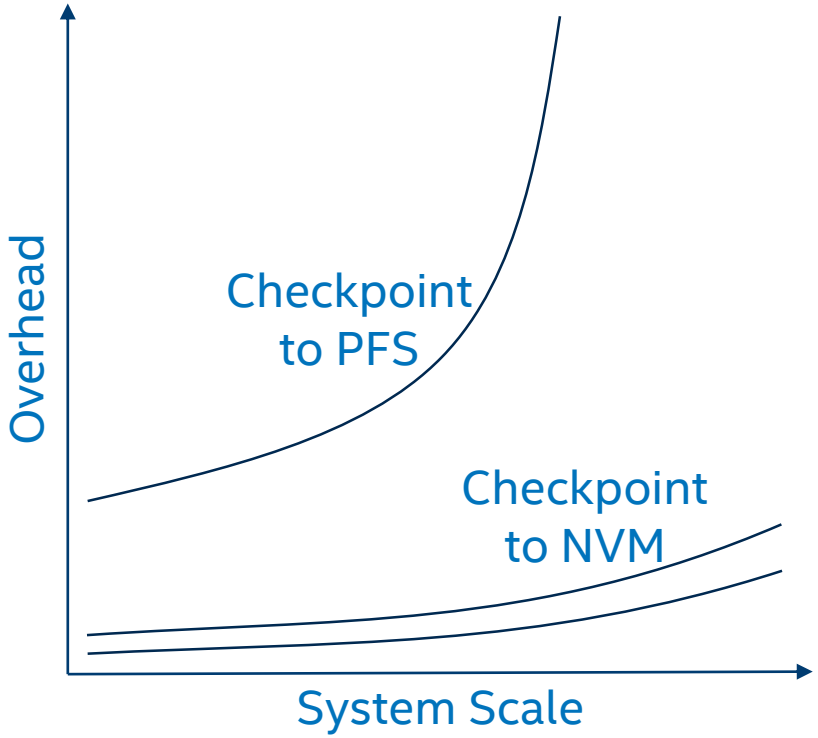
Focus on implementations and tools

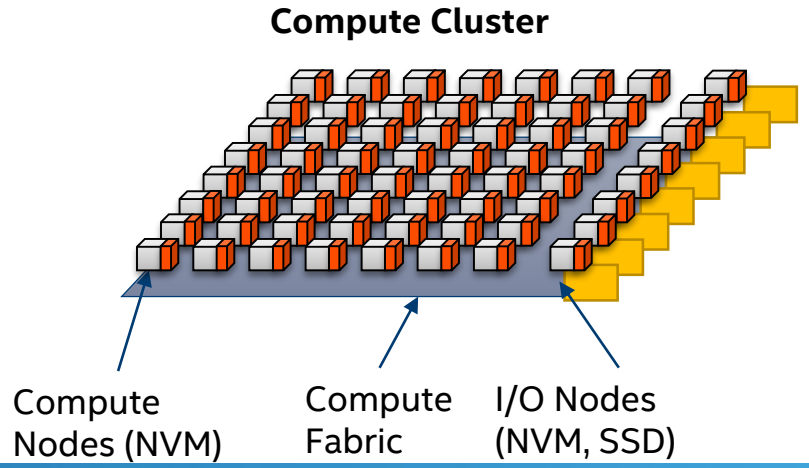
# CHECKPOINT/RESTART

NVM greatly reduces checkpoint/restart overhead

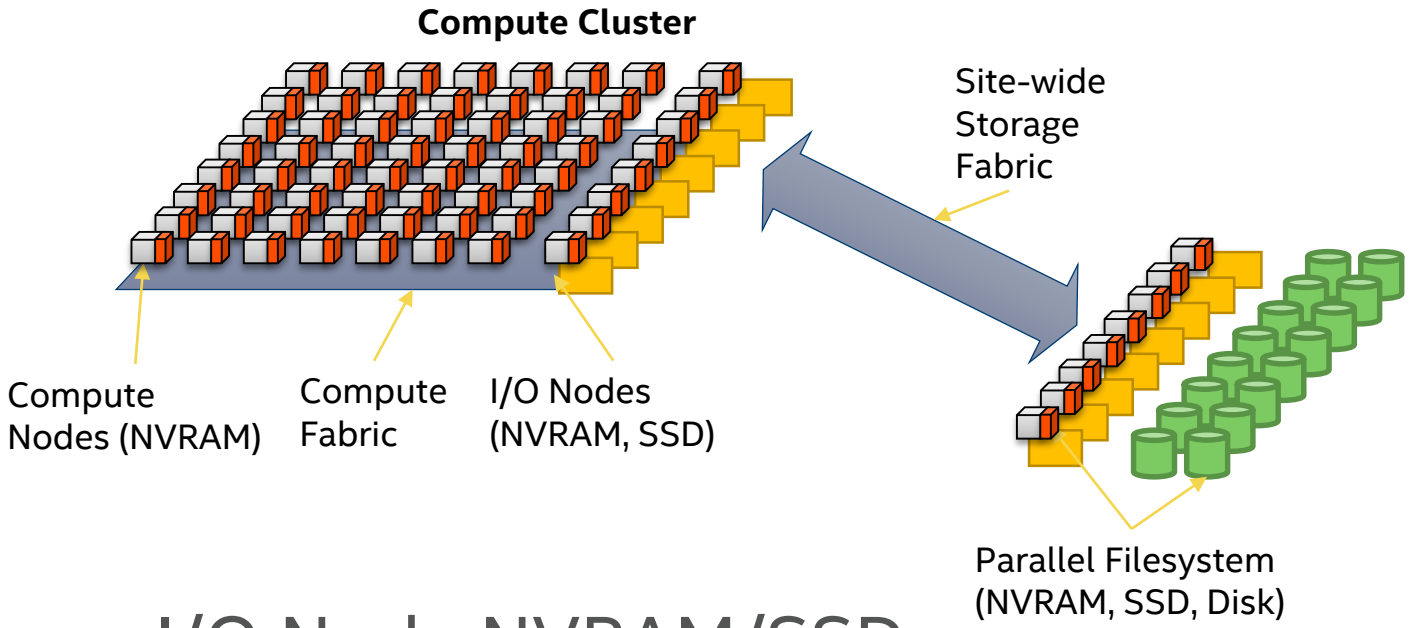
Where is the NVM located?

- Global (e.g., I/O Nodes)
  - Fabric BW
  - Globally accessible
  - Lower scale, resilience, …
- Local (i.e., Compute Nodes)
  - Full aggregate BW
  - Locality & durability concerns
- Opportunity for Local v. Global checkpoint
  - Local: HER + SER + OER
  - Global: HER



# STORAGE ARCHITECTURE




## Compute Node NVRAM

- Hot data
  - High valence & velocity
  - Brute-force, ad-hoc analysis
  - Extreme scale-out
- Full fabric bandwidth
  - O(1PB/s)→O(10PB/s)
- Extremely low fabric & NVRAM latency
  - Extreme fine grain
  - New programming models

## I/O Node NVRAM/SSD

- Semi-hot data / staging buffer
- Fractional fabric bandwidth
  - O(10TB/s)→O(100TB/s)

## Parallel Filesystem NVRAM/SSD/Disk

- Site-wide shared warm storage
  - SAN limited – O(1TB/s)→O(10TB/s)
- Indexed data

# PERSISTENT MEMORY AND RESOURCE MANAGEMENT

Persistent memory complicates resource management

- Data locality introduces hysteresis

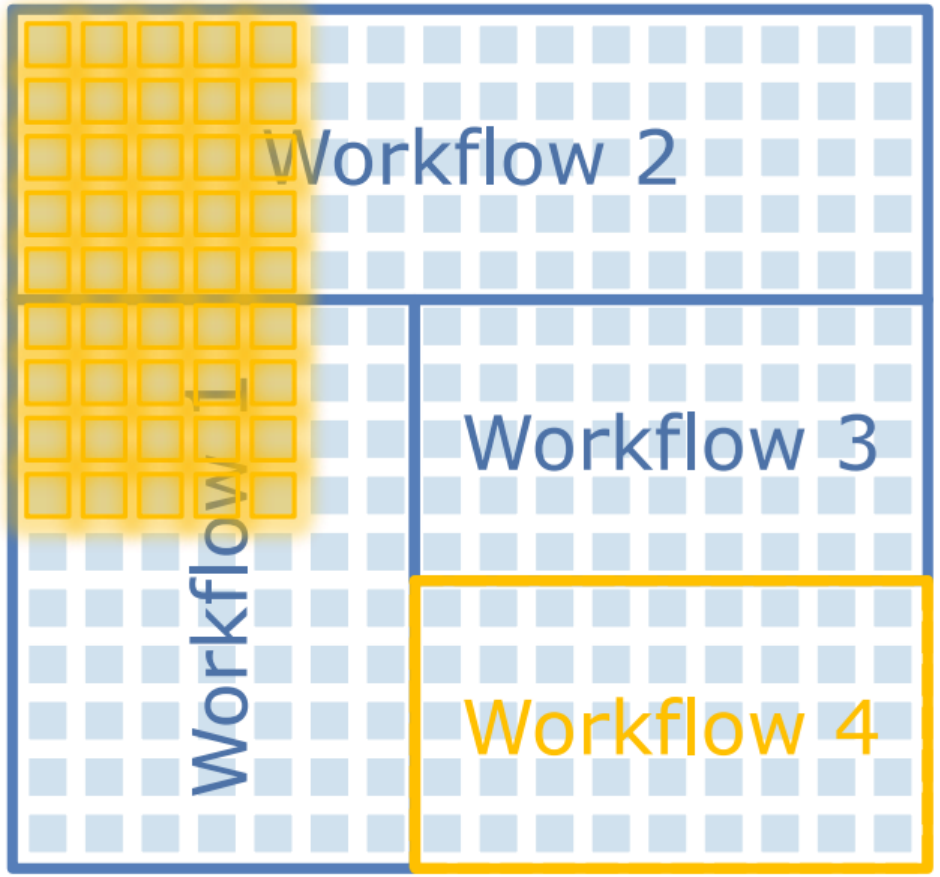
Data migration issues

- Space at destination
- Fabric interference

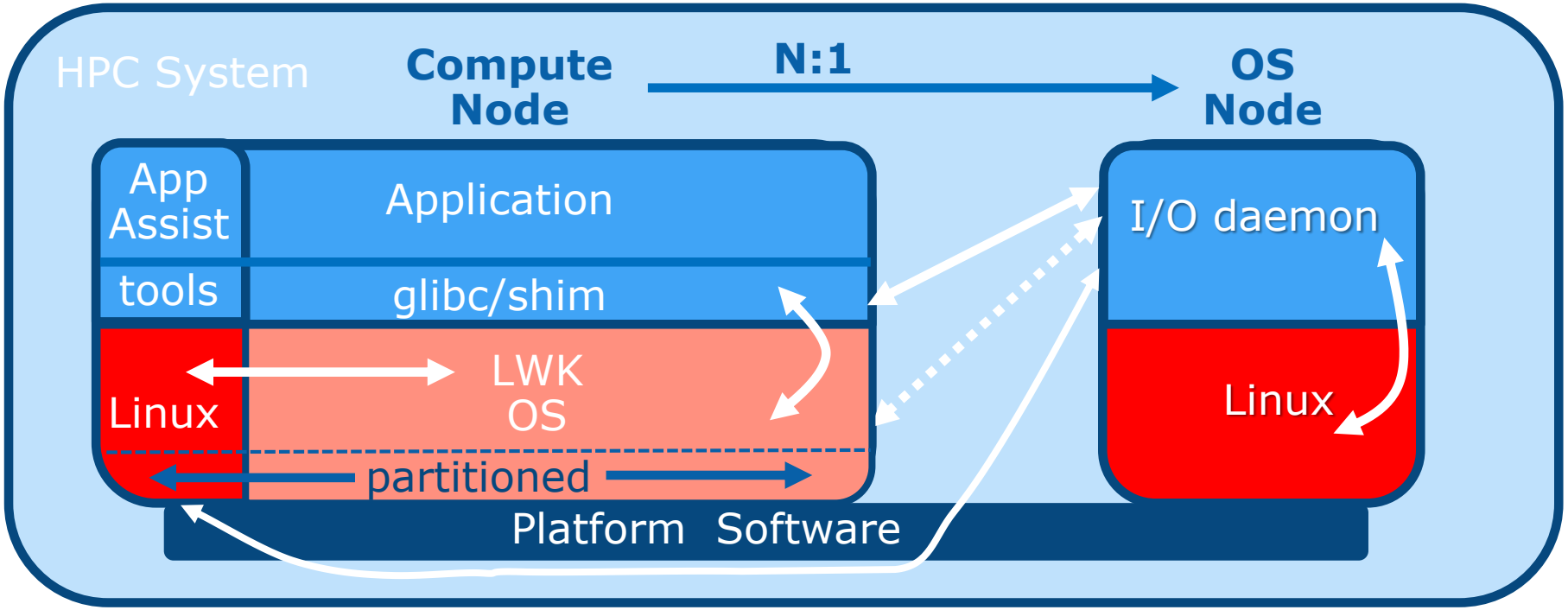
Data movement v. data reconstruction

Security considerations

- Encryption may be useful for some



# SCALABLE OS




- CNOS that fully supports Linux API and ABI
- Nimble to support new technology effectively
- Move to hierarchy of OS offload for scalability
- Support fine-grained threading and asynchronous requests
- Provide support for and be amenable to running on differentiated cores

# BACKUP

# APPLICATION SCALABILITY

Substantial pressure on application scalability

- System scale
- Cores/threads

MPI + X

- OpenMP
- C++ (lambda), RAJA, Kokkos, …
- OCR, AutoOCR, CnC, …

http://openmp.llvm.org/
https://www.openmprtl.org/

https://01.org/open-community-runtime