The Impact of Scalable Tool Infrastructure on Systems with 3D Mesh Networks

Philip C. Roth
Future Technologies Group
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831
rothpc@ornl.gov

OAK RIDGE
National Laboratory

High-End Computing (HEC) System Trends

- Increasing system component counts
  - 10Ks sockets today
  - Millions of cores soon

- At scalability limits of bandwidth-constant network topologies (e.g., fat trees)
  - Physical (form factor, bandwidth)
  - Expense

⇒ Increasing use of 3D mesh interconnection networks in large-scale systems

⇒ Continued use of batch queues

⇒ Limited access, controlled by allocations

HEC Platform Example 1: Cray XT

- Compute Processing Elements (PEs), Service PEs
- Cray SeaStar router, connected as 3D mesh with torus links possible
- Routes fixed at boot time; basically “x, then y, then z” routing

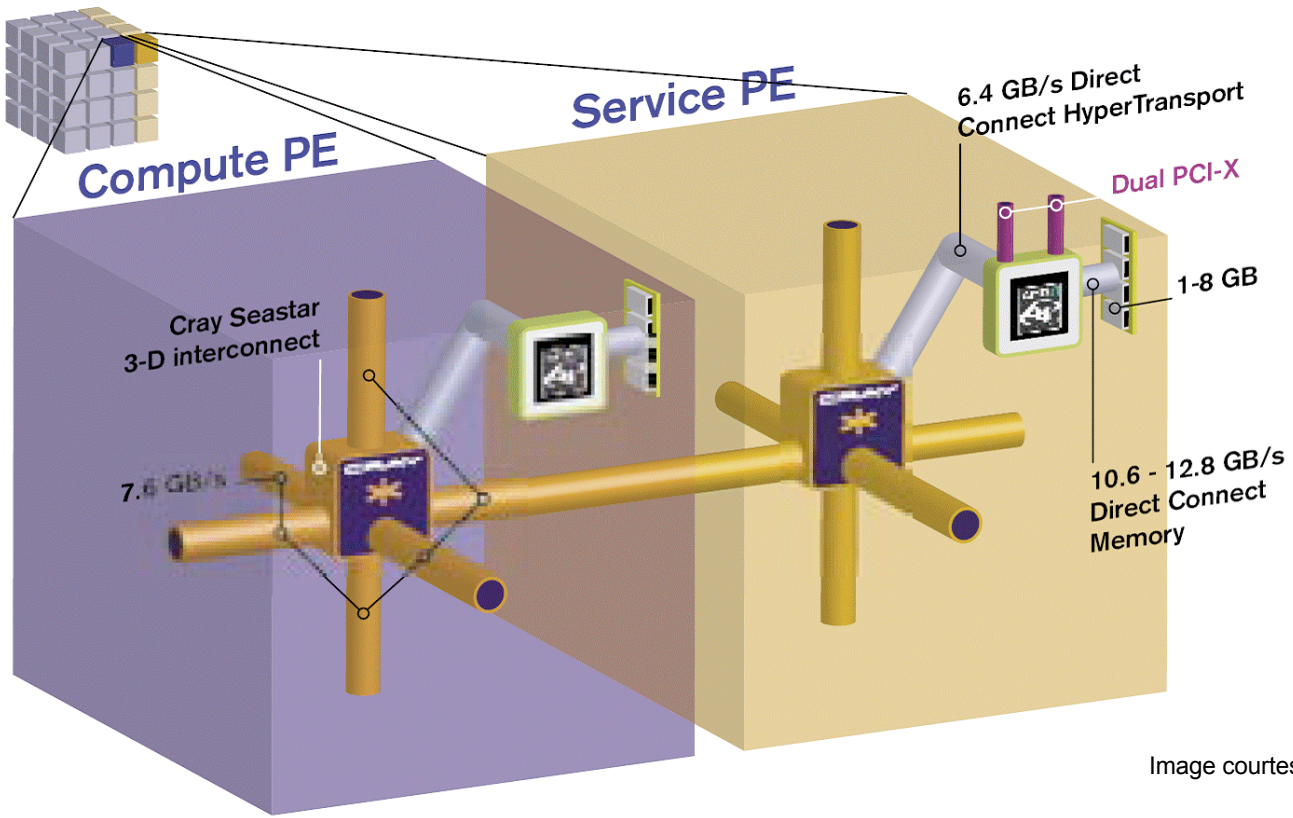


Image courtesy Cray, Inc.

HEC Platform Example 2: IBM Blue Gene

- Compute nodes, I/O nodes
  - Fixed ratio of compute nodes to I/O nodes
  - Fixed association of compute nodes with an I/O node
- Five networks, including 3D torus

HEC System Trends: Implications for Tools

- Emphasis on on-line, automated tools (e.g., recent Software Development Tools for Petascale Computing, Snowbird meetings)

- Emphasis on scalable performance, correctness, and system administration tools

⇒Increasing interest in scalable tool infrastructure, e.g. Tree-Based Overlay Networks (TBONs)

⇒Tools must thrive in HEC batch environment

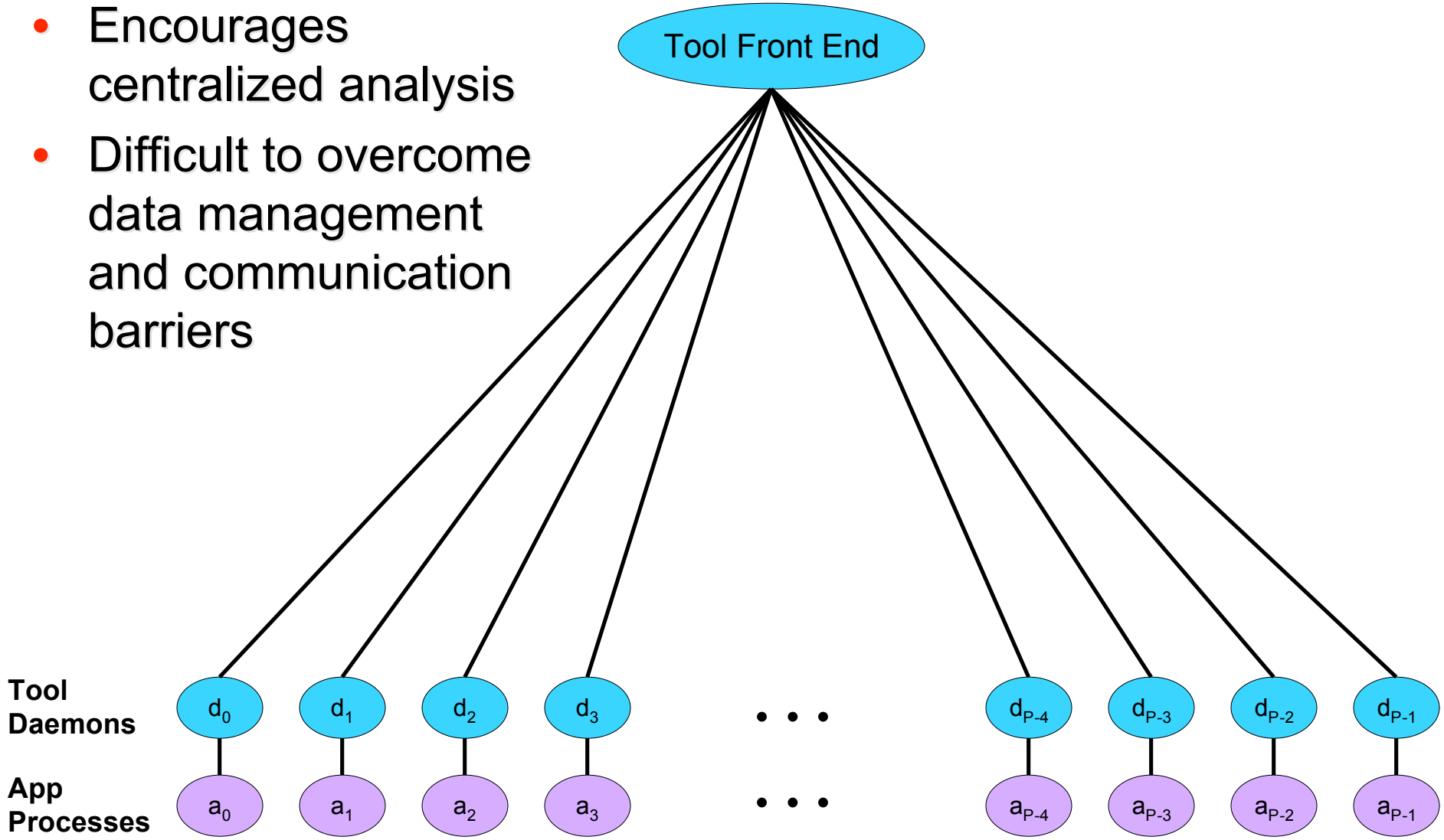
- Resources fixed at job submission time
- Minimal footprint

Tree-Based Overlay Networks (TBONs)

- Parallel tool infrastructure providing:
  - Scalable multicast
  - Scalable data synchronization and transformation
- Useful for parallelizing and distributing tool activities
  - Reduce latency
  - Reduce computation and communication load at tool front-end
- MRNet: TBON implementation with Barton Miller and Dorian Arnold (University of Wisconsin); See http://www.paradyn.org/mrnet

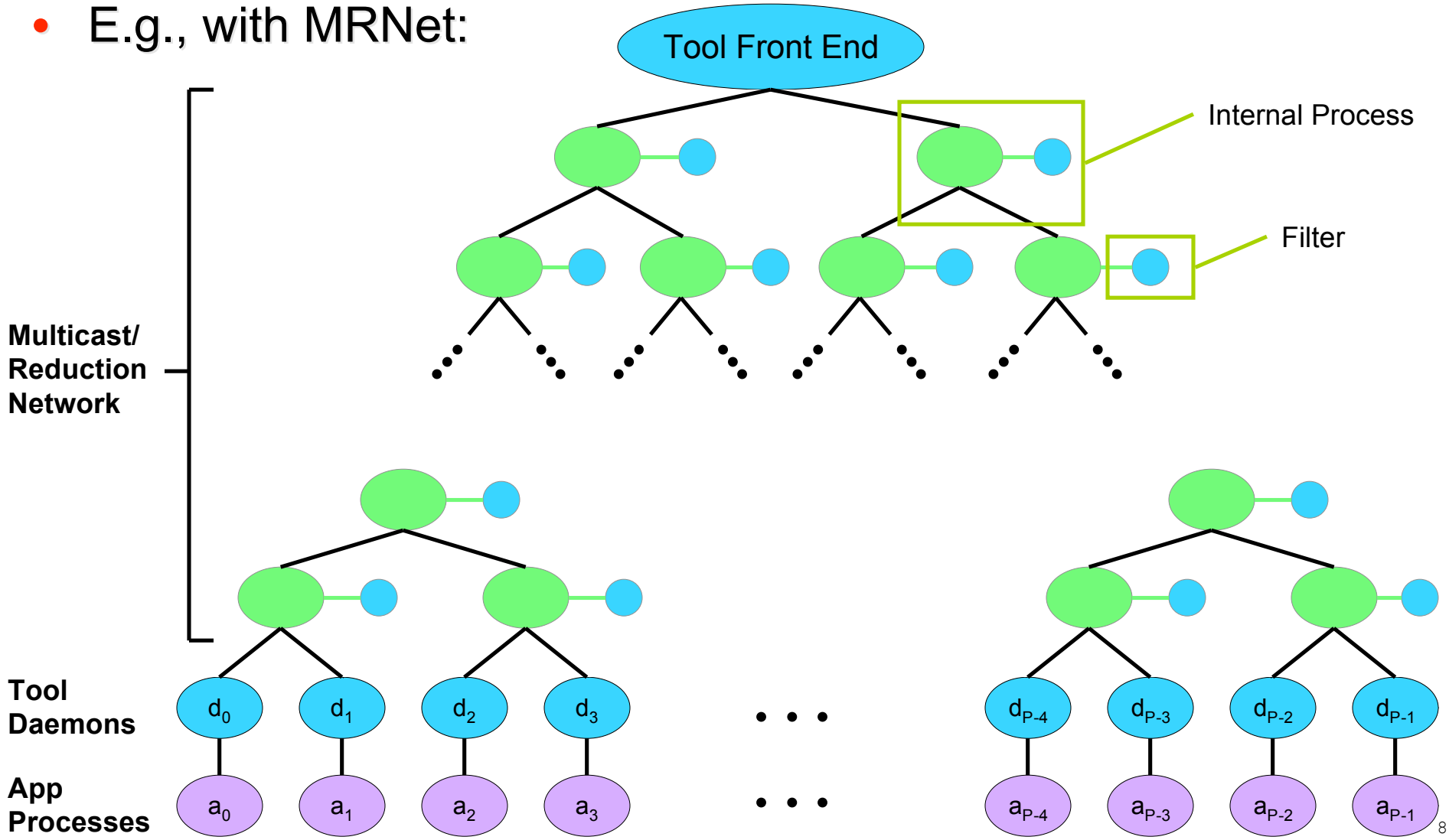
Traditional Tool Organization

- Encourages centralized analysis
- Difficult to overcome data management and communication barriers



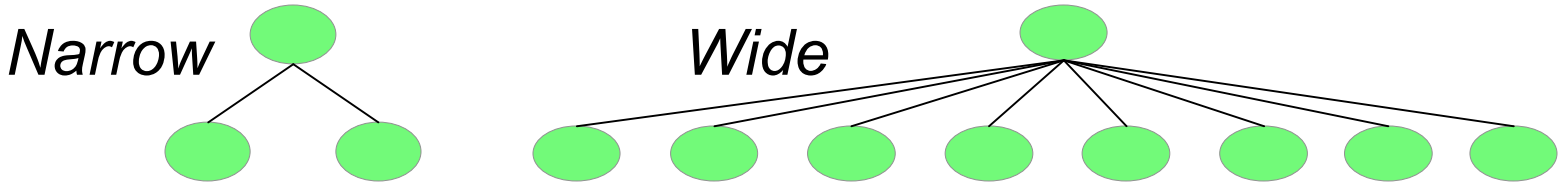
TBON-Based Tool Organization

- Network of processes between tool front-end and back-end
- E.g., with MRNet:

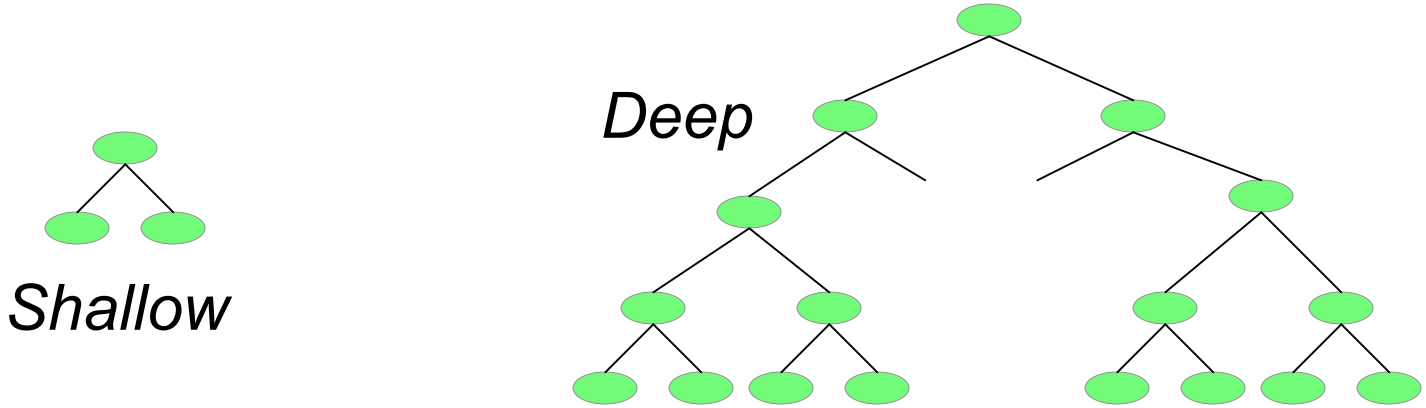


TBON Configuration Flexibility

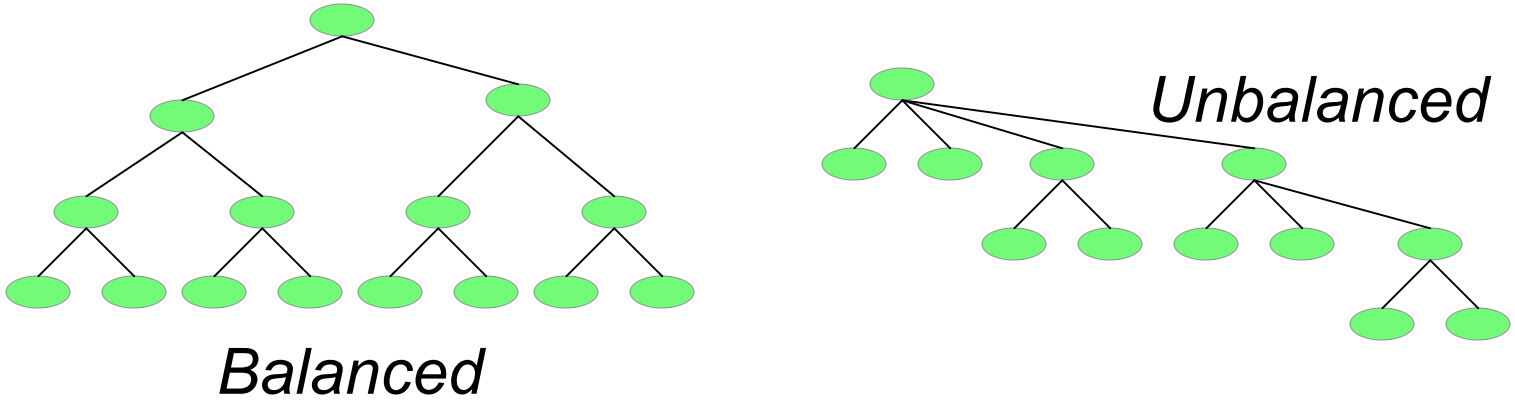
Fanout:



Depth:



Balance:



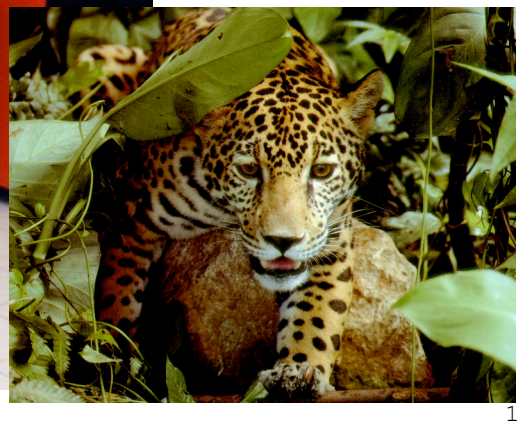
TBONs and HEC 3D Mesh Systems

"Fixed resources, minimal tool footprint…"

- Where to place tool and TBON processes on the system?
  - Service nodes
  - Compute nodes
  - External resources
- Which TBON configuration?
  - How many TBON processes?
  - Which TBON connection topology?

Context: Jaguar

- Hybrid Cray XT3/XT4 system at Oak Ridge National Laboratory
- Full size: 3D torus network, 31 × 16 × 24
- Currently: 23 × 16 × 24
  - Catamount on compute nodes
  - 32 cabinets removed in transition to Compute Node Linux



XT Process Placement

- Non-dedicated service nodes
  - Placement on service nodes is undesirable
  - BG/L MRNet-based tool efforts (e.g. UW/LLNL STAT scalable tracing tool) place MRNet on service nodes

- Lack of support systems + modest inter-system bandwidth
  - Placement on external cluster is undesirable

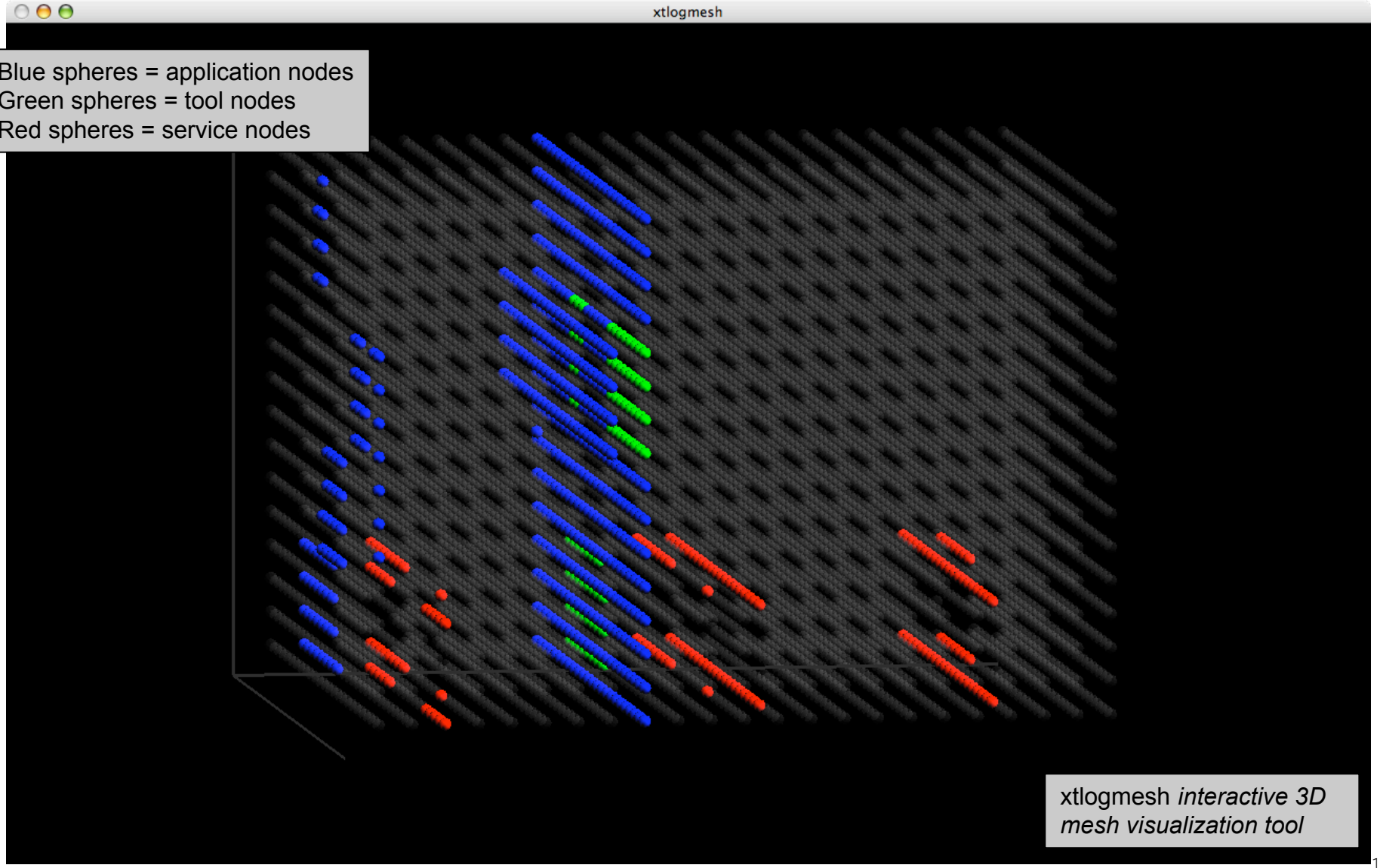
⇒Working assumption: tool processes run on compute nodes

Jaguar Placement Tests

- Used simple MPI/Portals program to collect node number and position within the XT mesh

- Batch job launched two *independent* instances of the program:
  - 512 application nodes (1024 application processes in "virtual node" mode)
  - 72 tool nodes (enough for balanced 8-way TBON topology, assuming front-end is on batch script or login node)

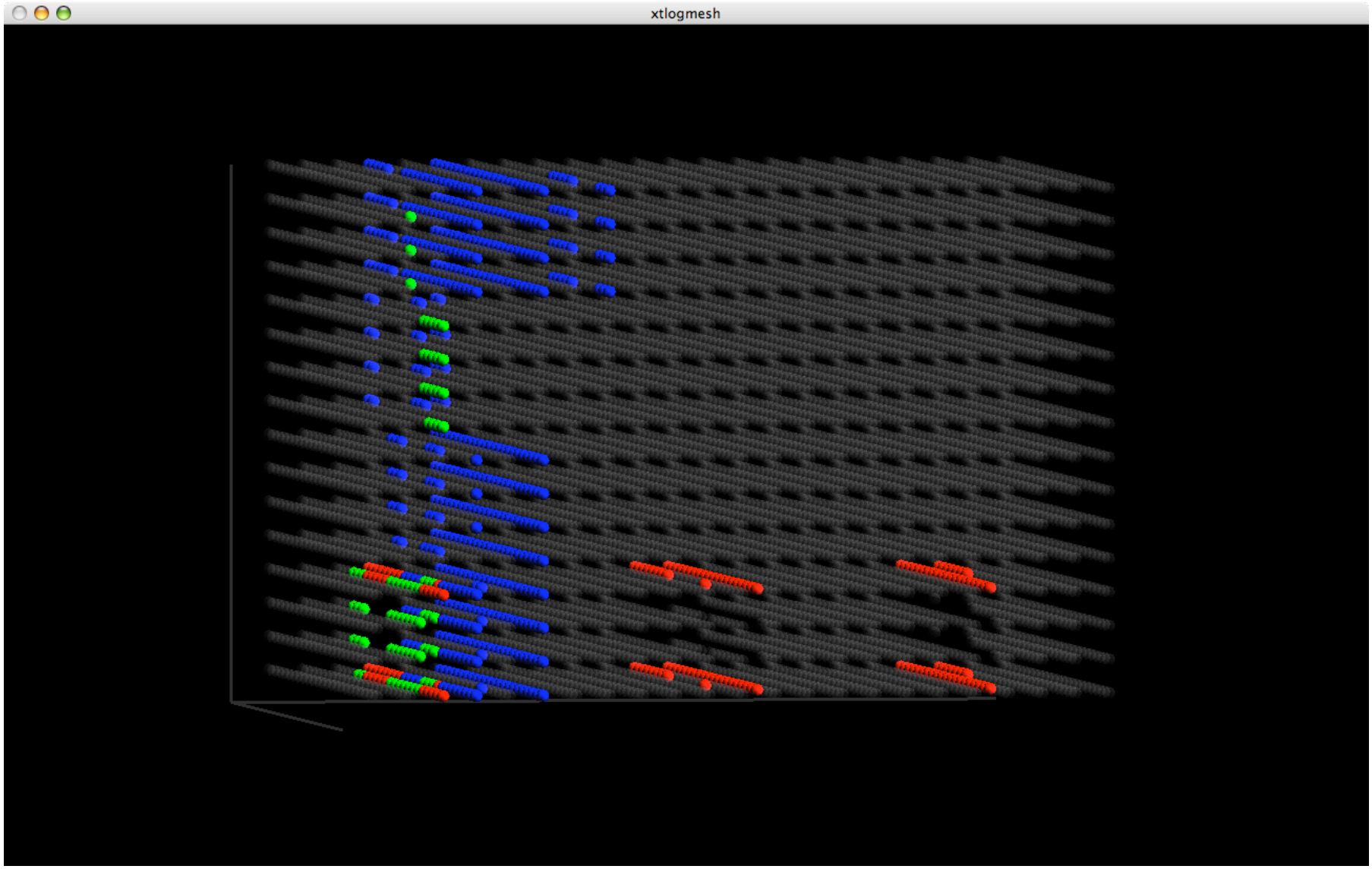
Jaguar Placement Trial 1



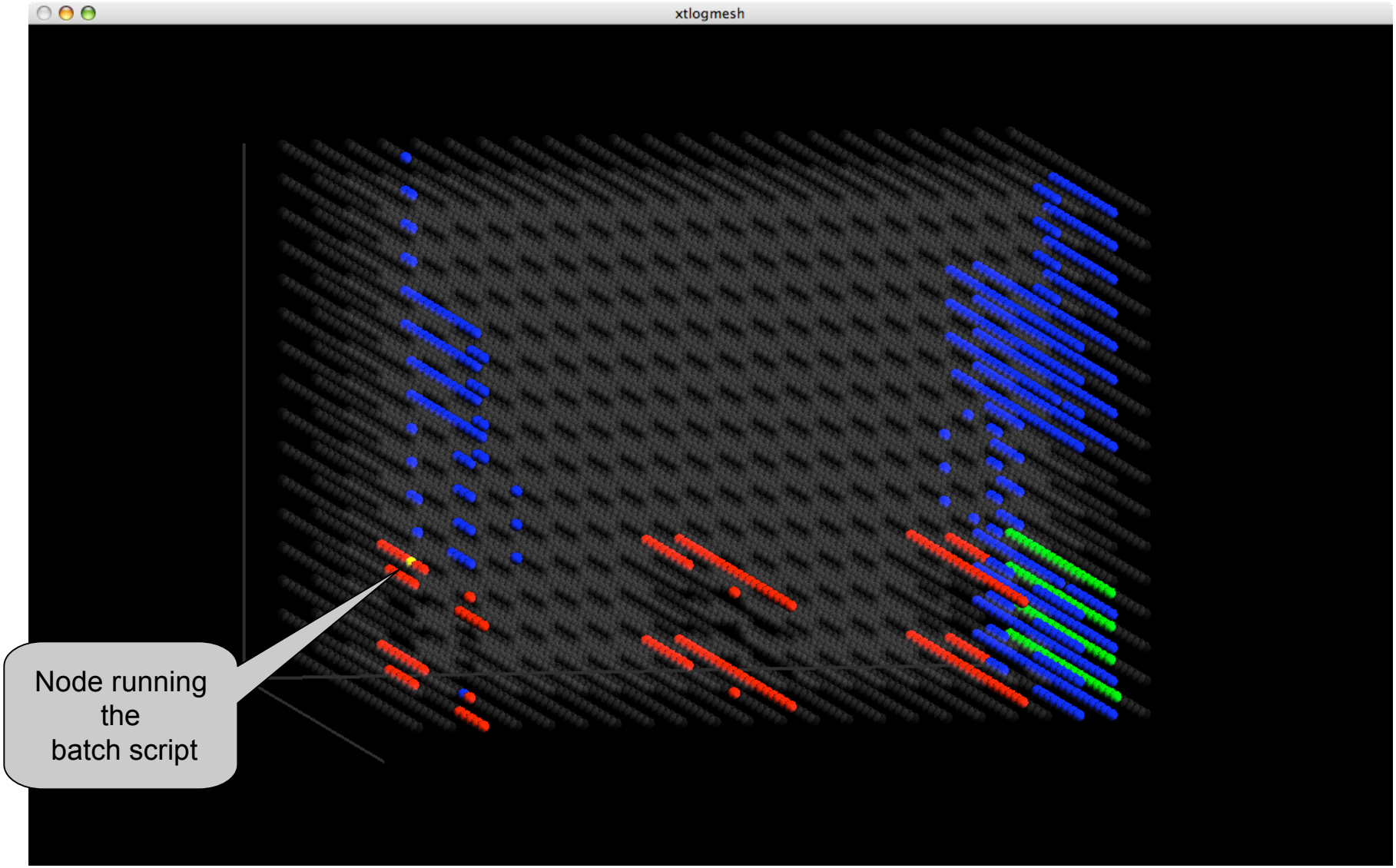
Blue spheres = application nodes
Green spheres = tool nodes
Red spheres = service nodes

xtlogmesh *interactive 3D mesh visualization tool*

Jaguar Placement Trial 2



Jaguar Placement Trial 3



XT Process Placement

- Space partitioning causes system fragmentation over time
  - Likely to get discontiguous nodes for large jobs
  - Not likely to get same “shape” across runs
  - Blue Gene’s large allocation blocks trade this against “internal fragementation”

- App process vs. tool process placement
  - Can detect nodes allocated to batch job
  - Can use job launcher to start independent parallel programs on targeted sets of nodes
  - Both Catamount and Compute Node Linux (details differ)

Resource Usage Impact

- Time on large HEC systems controlled by allocations
- Want to maximize the amount of work done using the allocation

- Often a time dilation from using a tool against a running parallel program
- With TBON processes on compute nodes, also a *space* dilation charged against the allocation

Example Usage Model: Baseline Case

- Simulation in strong scaling mode (fixed problem size)
- Run by itself (without a tool)

- A is the allocation (node-hours)
- N is the number of nodes per run
- R is simulation throughput (timesteps per hour)
- T is time per run (hours)

- Then number of timesteps per allocation is

$$\frac{AR}{N}$$

Example Usage Model: With Background Tool

- D_t (>1) is time dilation factor from running program with a background tool
  - Always running
  - No benefit to program performance
  - E.g., a health monitoring tool
- Number of timesteps per allocation is

$$\frac{AR}{ND_t}$$

- With space dilation Ds (also >1), number of timesteps per allocation is

$$\frac{AR}{ND_sD_t}$$

Usage Model: Complex Performance Tool Example

- Hopefully tools give insight to improve performance
- Assume usage scenario where the following is repeated:
  - Program is run k times without a tool
  - Program is run once under a tool
  - Results are analyzed, the program optimized, and the runtime improves by a factor of f (I.e., $T_{old} = fT_{new}$)
- Then the number of timesteps performed using the allocation is

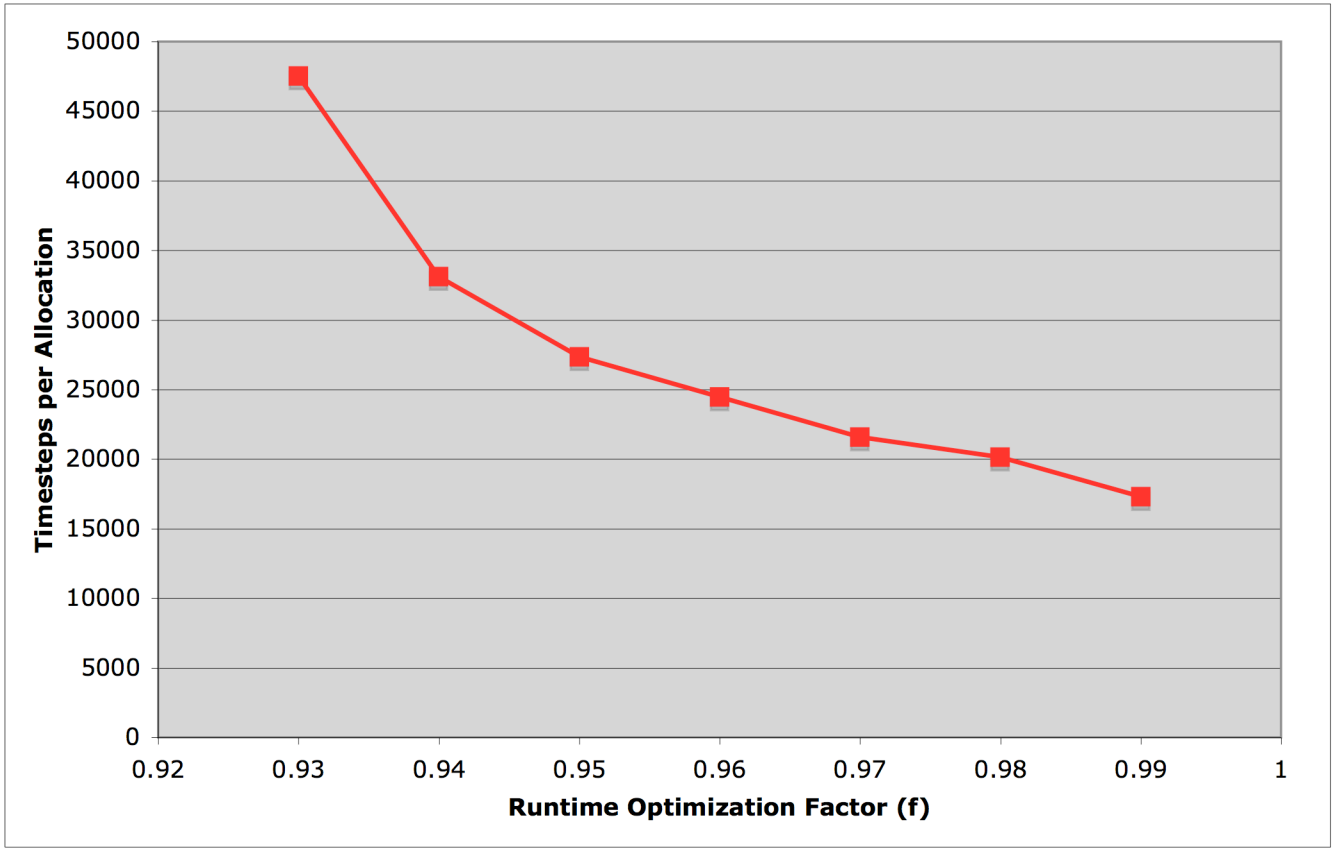
$$mRT(k+1)$$

- Where m is the largest integer such that

$$NT(k + D_s D_t)\frac{1 - f^{m+1}}{1 - f}$$

Timesteps vs. Optimization Factor

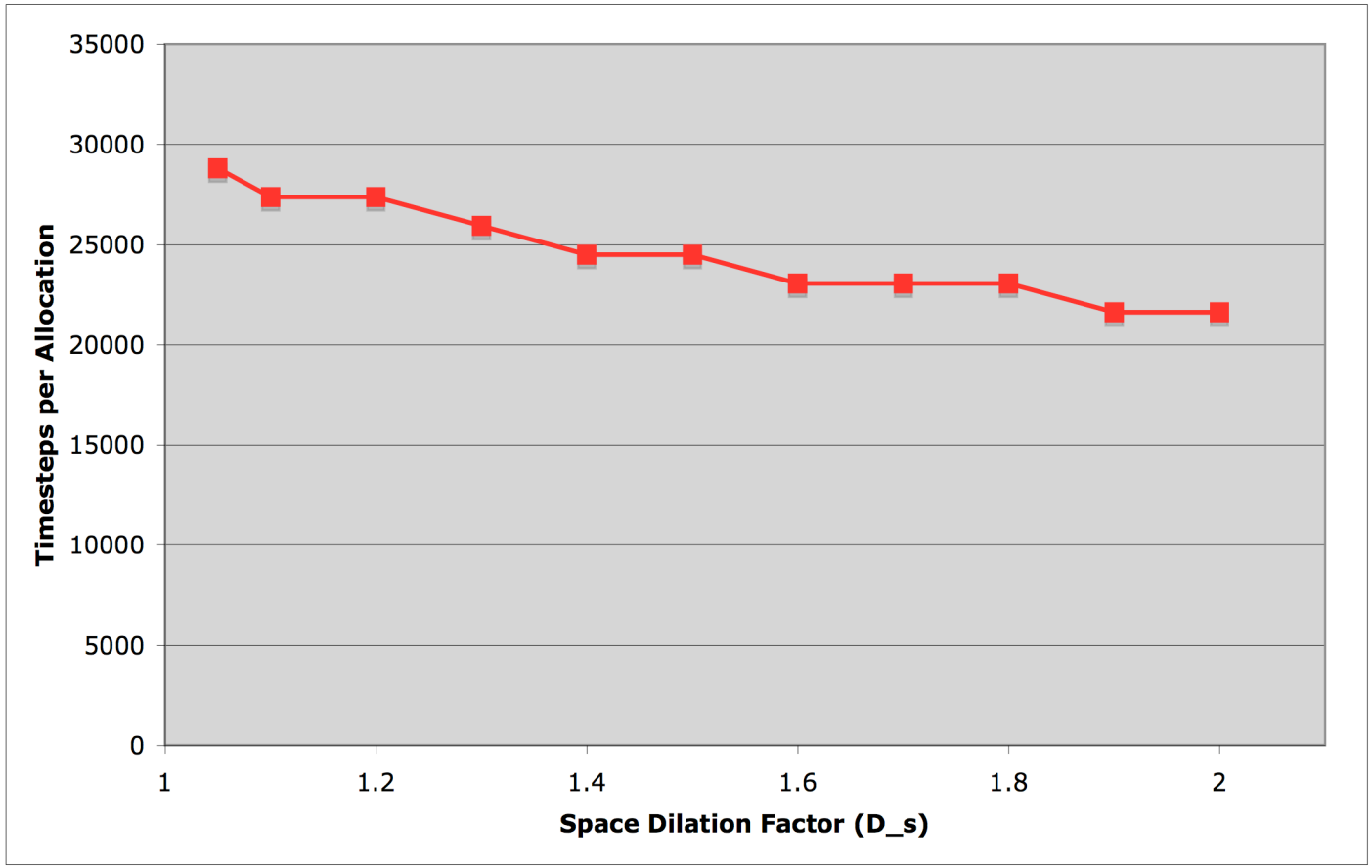
- *A=1M node-hours*
- *N=1024*
- *D_s = 1.1*
- *D_t = 1.1*
- *R = 20*
- *T = 12*
- *K = 5*



With the speculated optimization model, the number of timesteps is over twice as many if can optimize away 7% of the runtime compared to 2% of runtime

Timesteps vs. Space Dilation

- *A=1M node-hours*
- *N=1024*
- *f = 0.95*
- *D_t = 1.1*
- *R = 20*
- *T = 12*
- *K = 5*



Ongoing Work

- Performance prediction and validation
  - Modeling
  - Simulation
- Refining “allocation usage” model
- MRNet implementation for XT using XT’s Portals data communication layer
  - One-sided operations, OS- and application-bypass differ from original TCP/IP communication approach
  - For model/simulation validation, tool development

Acknowledgements

- Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U. S. Department of Energy under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

Summary

- Systems with 3D mesh (torus) networks and batch queues will remain prominent in HEC landscape
- TBON scalable tool infrastructure must be mapped carefully within allocated nodes on some systems
  - Both time and space dilation
  - Impacts achievable tool performance
  - Impacts application network traffic (both application under study and others')
  - Using compute nodes for TBONs can greatly impact ability to do science, both positively and negatively

Perspectives and Provocation

- A sobering perspective
  - *Extra nodes? I always assumed I didn't have any extra resources to work with. (del Signore, TotalView Technologies)*

- Is this "wasting" compute nodes? Does the answer change if they are used instead for fast checkpoint?
- Tool/Application codesign? (Even *a la* "Google"?)

- General information:
  - rothpc@ornl.gov
  - http://ft.ornl.gov

- MRNet:
  - paradyn@cs.wisc.edu
  - http://www.paradyn.org/mrnet