Office of Science
U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

The University of Reading

Middleware in Modern High Performance Computing System Architectures

Christian Engelmann[1,2], Hong Ong[1],
Stephen L. Scott[1]

[1] Oak Ridge National Laboratory, Oak Ridge, USA
[2] The University of Reading, Reading, UK

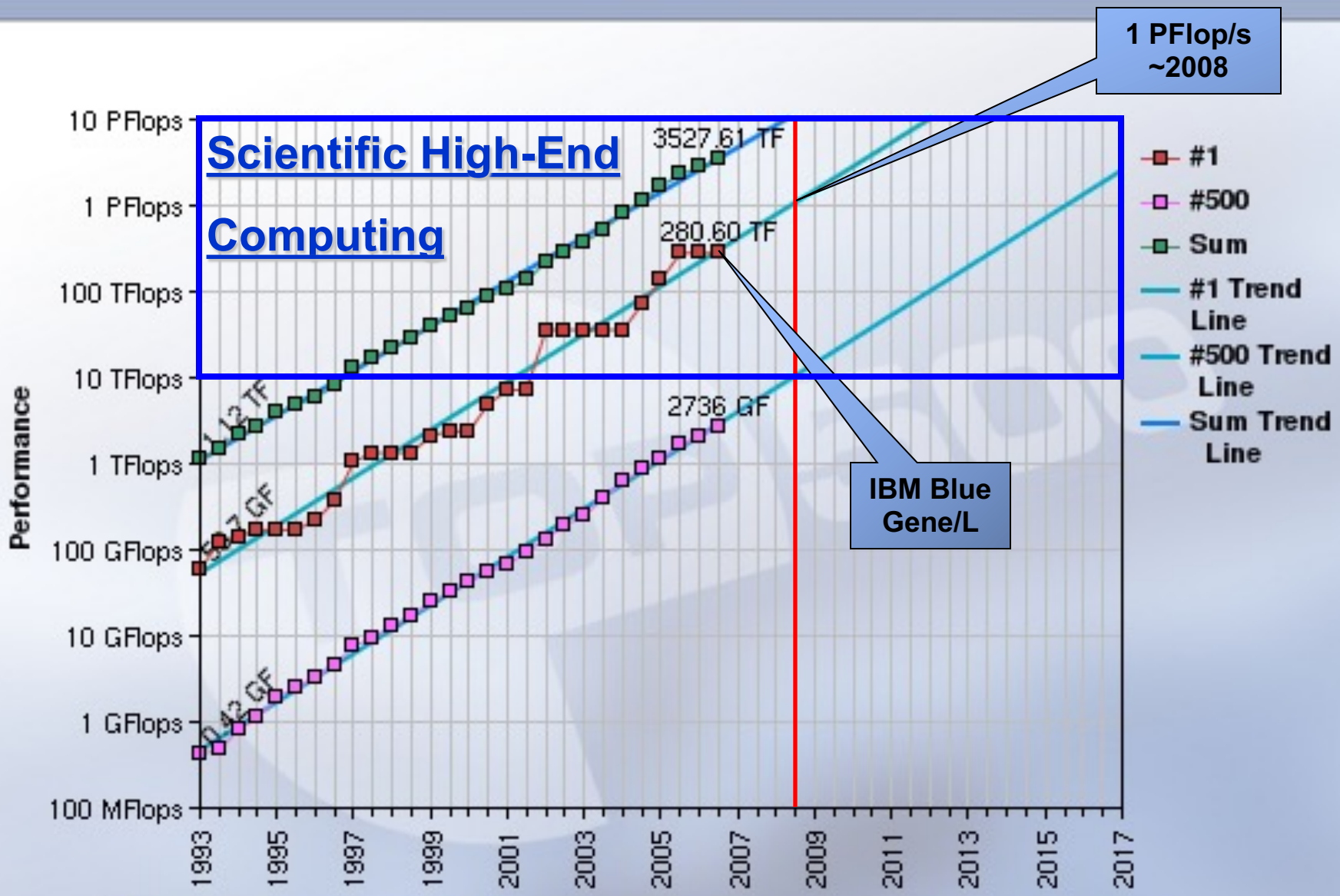
Talk Outline

- Scientific high-end computing (HEC)
- Trends in HPC system architectures
- Trends in HPC middleware architectures
- Modern HPC middleware
- The multi-core age: HPC for everyone

Scientific High-End Computing (HEC)

- Large-scale HPC systems.
  - Tens-to-hundreds of thousands of processors.
  - Current systems: IBM Blue Gene/L and Cray XT4
  - Next-generation: petascale IBM Blue Gene and Cray XT
- Computationally and data intensive applications.
  - 10 TFLOP – 1PFLOP with 10 TB – 1 PB of data.
  - Climate change, nuclear astrophysics, fusion energy, materials sciences, biology, nanotechnology, …
- Capability vs. capacity computing
  - Single jobs occupy large-scale high-performance computing systems for weeks and months at a time.

Projected Performance Development

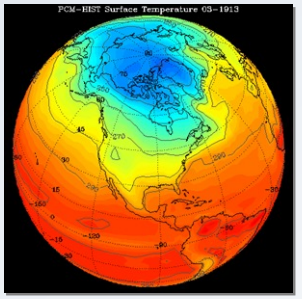


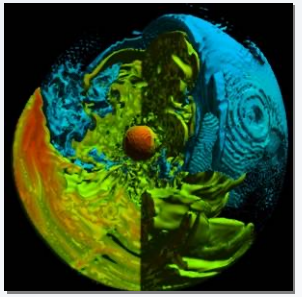
National Center for Computational Sciences

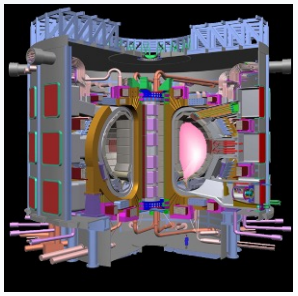
- 40,000 ft^2 (3700 m^2) computer center:
  - 36-in (~1m) raised floor, 18 ft (5.5 m) deck-to-deck
  - 12 MW of power with 4,800 t of redundant cooling
  - High-ceiling area for visualization lab:
    - 35 MPixel PowerWall, Access Grid, etc.

- 2 systems in the Top 500 List of Supercomputer Sites:
  - Jaguar: 10? Cray XT3, MPP with 11508 dual-core Processors ⇨ 119 TFlop.
  - Phoenix: 32? Cray X1E, Vector with 1014 Processors ⇨ 18 TFlop.

At Forefront in Scientific Computing and Simulation

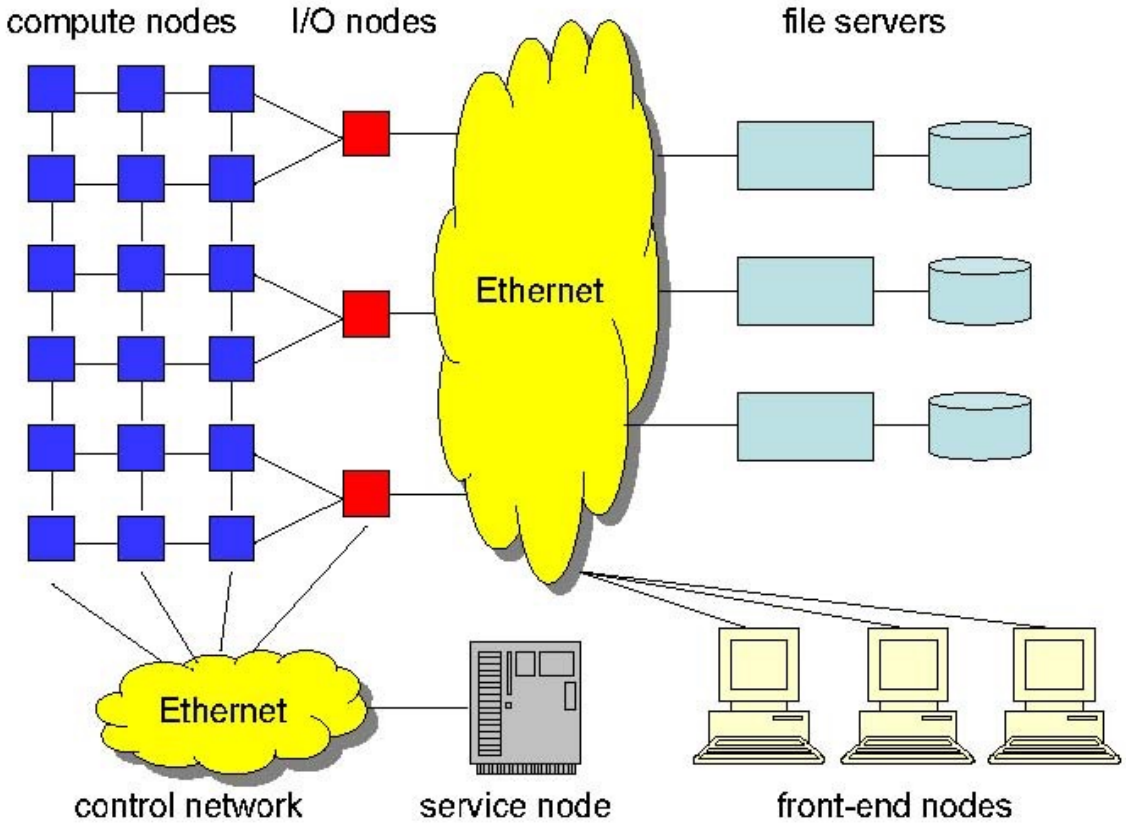






- Leading partnership in developing the National Leadership Computing Facility
  - Leadership-class scientific computing capability
  - 100 TFlop/s in 2007 (recently installed)
  - 250 TFlop/s in 2007/8 (commitment made)
  - 1 PFlop/s in 2008/9 (proposed)
- Attacking key computational challenges
  - Climate change
  - Nuclear astrophysics
  - Fusion energy
  - Materials sciences
  - Biology
- Providing access to computational resources through high-speed networking (10Gbps)

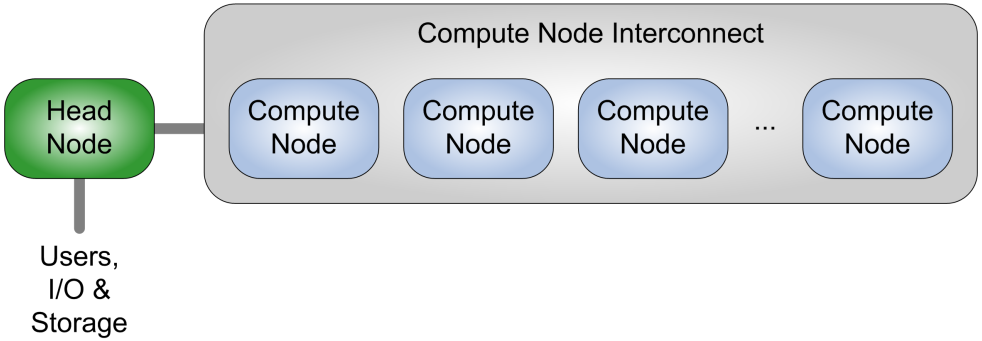
Typical HEC System Architecture



- Compute nodes (10,000+)
- Front-end, service, and I/O nodes (50+)

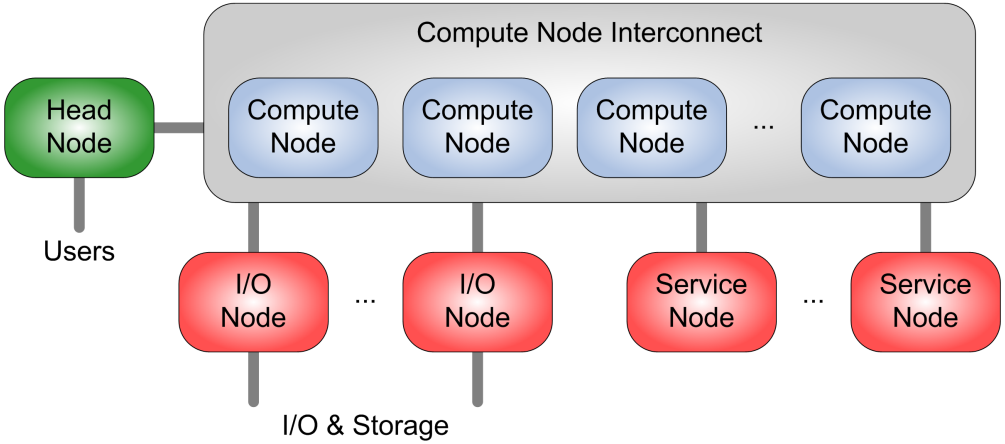
Image source: Moreira et al., "Designing a Highly-Scalable Operating System: The Blue Gene/L Story" *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, Nov. 11-17, Tampa, FL, USA.*

Traditional Beowulf Cluster Architecture



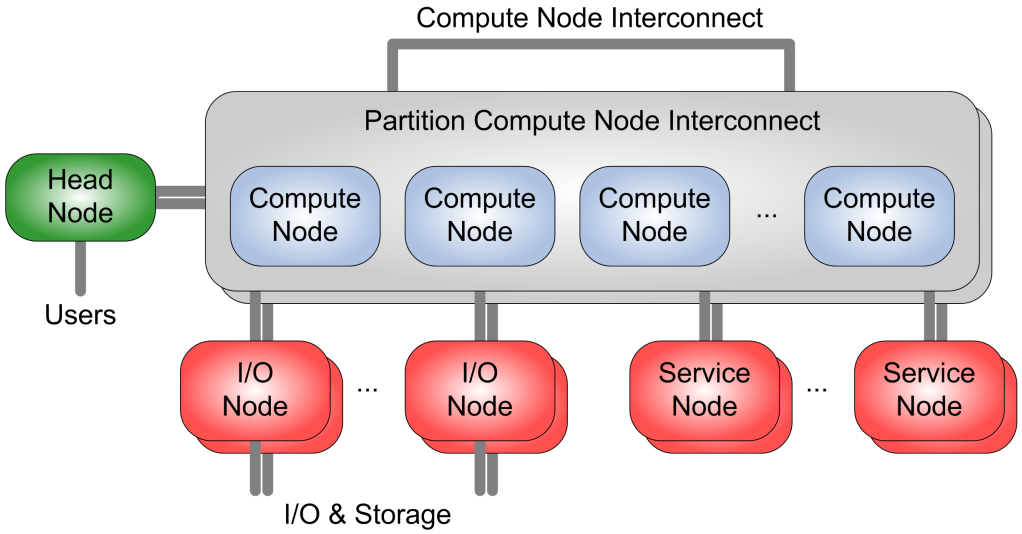
- Single head node manages entire HPC system
- System-wide services are provided by head node
- Local services are provided by compute nodes
- Full ("fat") operating system on compute nodes
  - Operating system kernel (kernel, kernel daemons and modules)
  - Operating system services (daemons and libraries)
  - Middleware services (daemons and libraries)

Modern HPC System Architecture



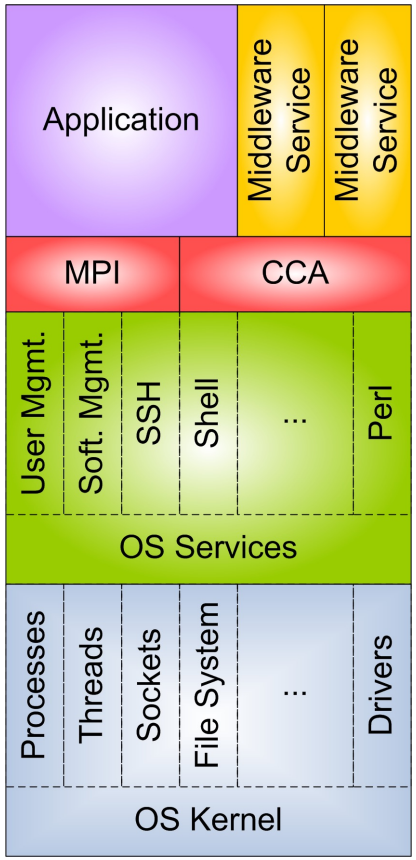
- Single head node and additional service node manage entire HPC system
- System-wide services are provided by head node and are offloaded to service nodes
- Local services are provided by service nodes and compute nodes
- Lightweight ("lean") operating system on compute nodes
  - Operating system kernel (kernel) – **kernel daemons do not exist**
  - Operating system services (libraries) – **daemons are on service nodes**
  - Middleware services (libraries) – **daemons and some libraries are on service nodes**

Modern Partitioned HPC System Architecture

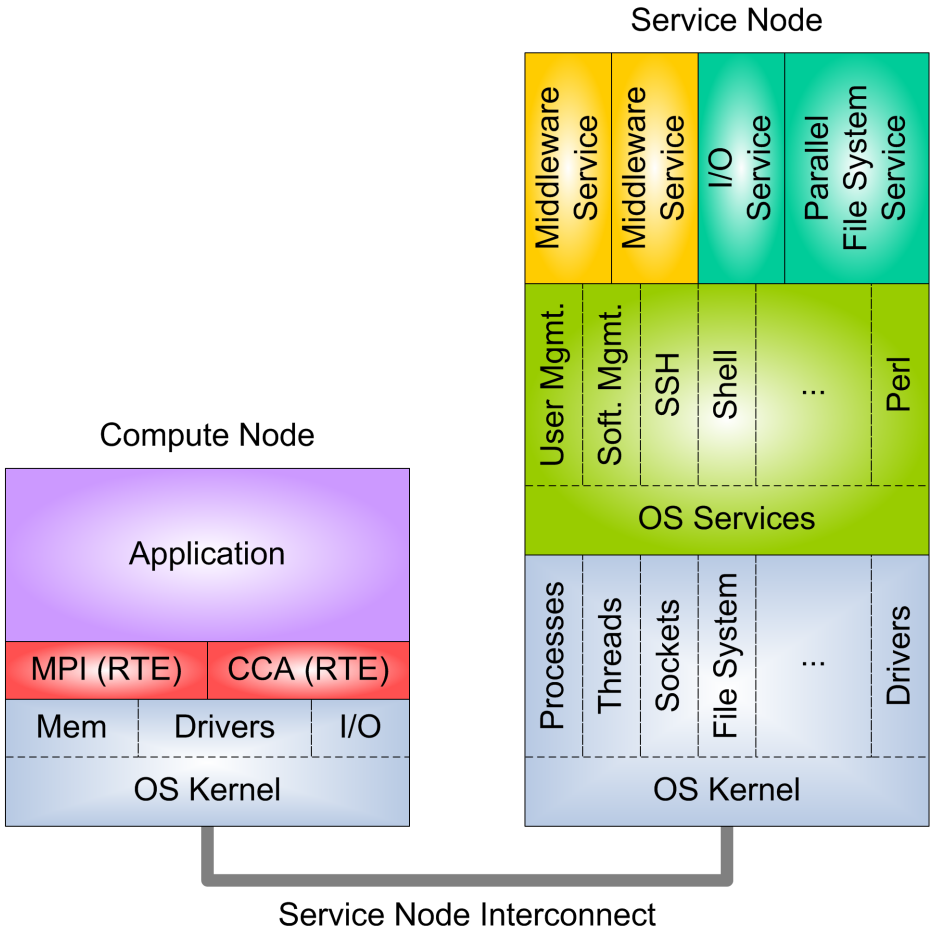


- Single head node manages entire HPC system
- Service nodes manage and support compute nodes belonging to their partitions
- **OS and middleware on compute nodes interact within partitions via service nodes**
- **OS support and middleware on service nodes interact across partitions**
- **Only system management services and message passing function transparently**

Traditional Compute Node Software Architecture



Modern Compute Node Software Architecture



HPC Middleware

- Provides certain basic services:
  - message passing layer
  - fault tolerance support
  - runtime reconfiguration
- Offers advanced services:
  - application steering mechanisms
  - user interaction techniques
  - scientific data management
- Each is typically an individual piece of software
- This has led to the **yet another library** and **yet another daemon** phenomenons

Modern HPC Middleware

- Employs lean compute nodes using lightweight operating systems in order to:
  - increase performance and scalability
  - reduce compute node software to the absolute necessary
- Only basic services are on compute nodes (if needed)
- Advanced and other basic services are supplied via service nodes using an RPC forwarding mechanism
- The lightweight operating system on compute nodes and the reliance on service nodes drastically change HPC middleware design and mechanisms.

Modern HPC Middleware Features

- Functionality:
  - Adaptation of HPC middleware software architecture is needed to the service node model
  - Delegation of responsibilities to service nodes is needed to interact across compute node partitions
- Performance and Scalability:
  - The RPC forwarding mechanism from compute nodes to service nodes incurs a latency and bandwidth penalty
  - Service nodes represent a bottleneck and a central point of control for the compute nodes they serve
  - Middleware service offload and load balancing techniques may be used to alleviate performance and scalability issues

Modern HPC Middleware Features

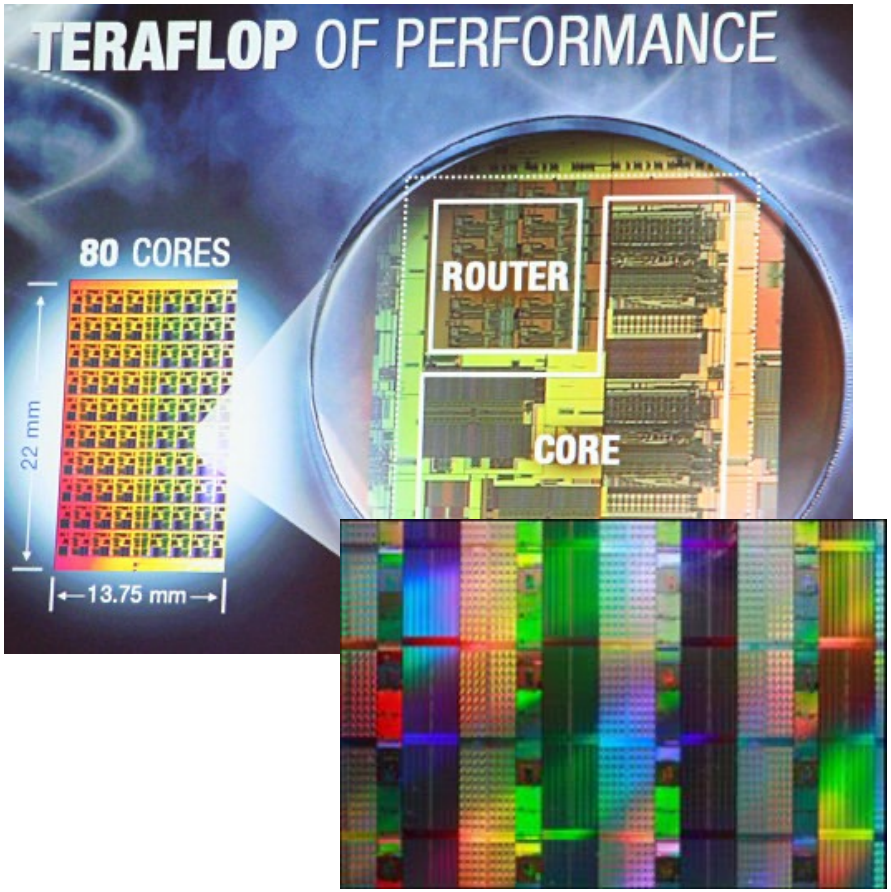
- Reliability:
  - Service nodes represent a central point of failure for the compute nodes they serve
  - Middleware service replication techniques may be used to improve reliability, availability, and serviceability (RAS)
- Slimming Down
  - Existing limitations of the lightweight OS on compute nodes, such as the missing dynamic linker
  - Existing features of the lightweight OS on compute nodes, like the RPC forwarding mechanism

Modern HPC Middleware Features

- Service-Oriented Middleware Architecture (SOA):
  - Bring an architectural advantage as we already know how to design and develop SOA middleware
  - Many existing solutions from the distributed systems community can be reused
  - Opportunity for integration with existing technologies:
    - Data stream processing on I/O service nodes for visualization
    - Interaction and application steering via service nodes
    - Service-level replication mechanisms for high availability
    - Service-level load balancing for QoS guarantees

In the Multi-Core Age, Modern HPC Middleware Architectures Will Affect Everyone

- As the number of cores on a chip increases, everyone will have a massively parallel HPC system.
- Lightweight operating systems and service oriented middleware will soon be on your desktop/laptop.
- Why do you think Microsoft hired Burton Smith (formerly Cray)?



MOLAR: Adaptive Runtime Support for High-end Computing Operating and Runtime Systems

- Addresses the challenges for operating and runtime systems to run large applications efficiently on future ultra-scale high-end computers.
- Part of the Forum to Address Scalable Technology for Runtime and Operating Systems (FAST-OS).
- MOLAR is a collaborative research effort (*www.fastos.org/molar*):

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Office of Science
U.S. DEPARTMENT OF ENERGY

NC STATE UNIVERSITY

OHIO STATE

LOUISIANA TECH UNIVERSITY

The University of Reading

CRAY THE SUPERCOMPUTER COMPANY

Office of Science
U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

The University of Reading

Middleware in Modern High Performance Computing System Architectures

Christian Engelmann[1,2], Hong Ong[1],
Stephen L. Scott[1]

[1] Oak Ridge National Laboratory, Oak Ridge, USA
[2] The University of Reading, Reading, UK