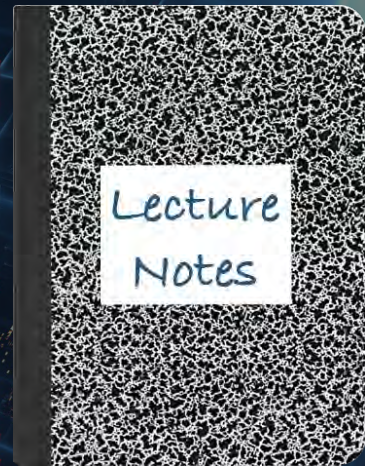


CS 417 – DISTRIBUTED SYSTEMS

Week 13: Infrastructure

Part 3: High Performance Computing (HPC) Clusters

Paul Krzyzanowski



© 2022 Paul Krzyzanowski. No part of this content may be reproduced or reposted in whole or in part in any manner without the permission of the copyright owner.

Supercomputers

2021's most powerful supercomputer: Fugaku 富岳 – Japan – developed by RIKEN & Fujitsu

- Performance
 - 442 petaflops
 - Fujitsu A64FX SoC processors: 7.6 million ARM cores
- OS
 - Compute nodes run McKernel (lightweight kernel designed for HPC) – a few hundred lines of C++
 - Communicate with I/O nodes that run Linux
- Communication
 - Torus Fusion (tofu) – proprietary interconnect developed by Fujitsu
 - 6-dimensional mesh/torus topology – full-duplex link with peak bandwidth of 5 GB/s in each direction
- Storage
 - 1.6 TB NVMe SSD for every 16 nodes
 - 150 PB shared storage – Lustre FS



[https://en.wikipedia.org/wiki/Fugaku_\(supercomputer\)](https://en.wikipedia.org/wiki/Fugaku_(supercomputer))

Supercomputers

2018's Most powerful supercomputer:

IBM AC922 – Summit at *Oak Ridge National Laboratory*

- 189 petaflops, >10PB memory
- 4,608 nodes
 - 6 NVIDIA Volta V100s GPUs } >27,000 GPUs
 - 2 IBM POWER9™ CPUs } >9,000 CPUs
 - 512 GB DDR4 + 96GB HBM2 RAM
 - 1600GB NV memory
 - 42 teraflops per node
- 100G InfiniBand interconnect
- 250 PB 2.5 TB/s file system
- OS: Red Hat Enterprise Linux
- Peak power consumption: 13 MW

See <https://www.olcf.ornl.gov/summit/>



- Supercomputers are *not* distributed computers
- Lots of processors connected by high-speed networks
- Shared memory access
- Shared operating system (all TOP500 run Linux)

Supercomputing clusters

- Supercomputing cluster
 - Build a supercomputer from commodity computers & networks
 - A distributed system
- Target complex, typically scientific, applications:
 - Large amounts of data
 - Lots of computation
 - Parallelizable application
- Many custom efforts
 - Typically Linux + message passing software + remote exec + remote monitoring



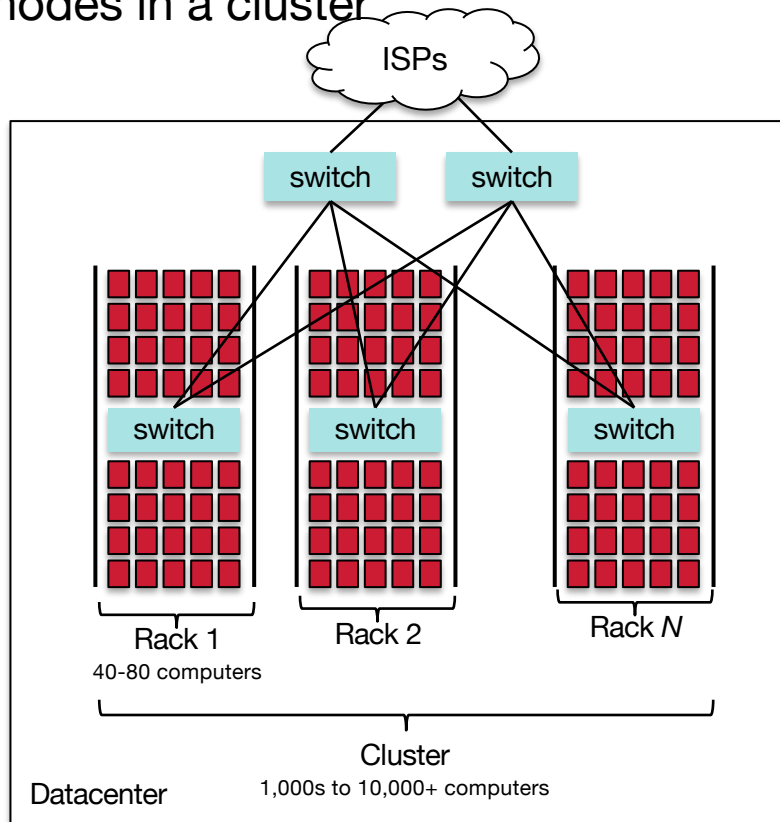
Cluster Interconnects

Cluster Interconnect

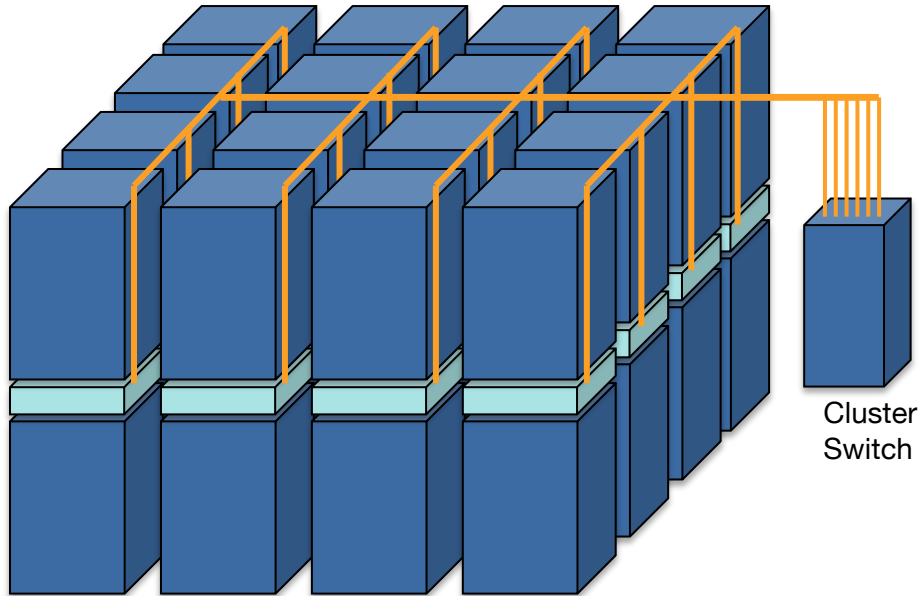
Provide communication between nodes in a cluster

Goals

- **Low latency**
 - Avoid OS overhead, layers of protocols, retransmission, etc.
- **High bandwidth**
 - High bandwidth, switched links
 - Avoid overhead of sharing traffic with non-cluster data
- **Low CPU overhead**
- **Low cost**
 - Cost usually matters if you're connecting thousands of machines
- Usually a LAN is used:
best **\$/performance** ratio



Cluster Interconnect



Cluster of 4x4 racks

Assume:

10 Gbps per server
40 servers per rack
⇒ 400 Gbps/rack

16 racks ⇒ 8 Tbps

Max switch capacity
currently ~ 5 Tbps
⇒ *Need at least two
cluster switches*

Switches add latency

Within one rack

- One **switch latency** $\approx <1...8 \mu\text{s}$ for a 10 Gbps switch
- Two links (to switch + from switch) @ 1-2 meters of cable
 - **Propagation time** in copper $\approx 2 \times 10^8 \text{ m/s} \approx 5 \text{ ns/m}$

Between racks in a cluster

- Three switch latency ($\approx <3...24 \mu\text{s}$)
- 4 links (to rack switch + to cluster switch + back to target rack)
- ~10-100 meters distance (50 ... 500 ns)

Add to that the normal latency of sending & receiving packets:

- System latency of processing the packet, OS mode switch, queuing the packet, copying data to the transceiver, ...
- **Serialization delay** = time to copy packet to media $\approx 1 \mu\text{s}$ for a 1KB packet on a 10 Gbps link

Dedicated cluster interconnects

TCP adds latency

- Operating system overhead, queueing, checksums, acknowledgements, congestion control, fragmentation & reassembly, ...
- Lots of interrupts
- Consumes time & CPU resources

How about using a high-speed LAN without the overhead?

- LAN dedicated for intra-cluster communication
 - Sometimes known as a **System Area Network (SAN)**
- Dedicated network for storage: **Storage Area Network (SAN)**

Example High-Speed Interconnects

Common traits

- **TCP/IP Offload Engines (TOE)**
 - TCP stack at the network card
- **Remote Direct Memory Access (RDMA)**
 - memory copy with no CPU involvement

Intel I/O Acceleration Technology (I/OAT) – combines TOE & RDMA

- Data copy without CPU, TCP packet coalescing, low-latency interrupts, ...



Example High-Speed Interconnects

Example: InfiniBand

- Switch-based point-to-point bidirectional serial links
- Link processors, I/O devices, and storage
- Each link has one device connected to it
- Enables data movement via **remote direct memory access** (RDMA)
 - No CPU involvement!
- Up to 250 Gbps/link
 - Links can be aggregated: up to 3000 Gbps with 12x links

IEEE 802.1 Data Center Bridging (DCB)

- Set of standards that extend Ethernet
- Lossless data center transport layer
 - Priority-based flow control, congestion notification, bandwidth management

Programming tools for HPC: PVM

PVM = Parallel Virtual Machine

- Software that emulates a general-purpose heterogeneous computing framework on interconnected computers
- Model: app = set of tasks
 - **Functional parallelism**: tasks based on function: input, solve, output
 - **Data parallelism**: tasks are the same but work on different data
- PVM presents library interfaces to:
 - Create tasks
 - Use global task IDs
 - Manage groups of tasks
 - Pass basic messages between tasks

Programming tools: MPI

MPI: Message Passing Interface

- API for sending/receiving messages
 - Optimizations for shared memory & NUMA
 - Group communication support
- Other features:
 - Scalable file I/O
 - Dynamic process management
 - Synchronization (barriers)
 - Combining results

HPC Cluster Example

Early example: Early (>20 years old!) effort on Linux – Beowulf

- Initially built to address problems associated with large data sets in Earth and Space Science applications
- From Center of Excellence in Space Data & Information Sciences (CESDIS)
 - Division of University Space Research Association at the Goddard Space Flight Center
- Still used!

This isn't one fixed package

- Just an example of putting tools together to create a supercomputer from commodity hardware

What makes it possible?

- Commodity off-the-shelf computers are cost effective
- Publicly available software:
 - Linux, GNU compilers & tools
 - MPI (message passing interface)
 - PVM (parallel virtual machine)
- Low cost, high speed networking
- Experience with parallel software
 - Difficult: solutions tend to be custom

What can you run?

- Programs that do not require fine-grain communication
- Basic properties
 - Nodes are dedicated to the cluster
 - Performance of nodes not subject to external factors
 - Interconnect network isolated from external network
 - Network load is determined only by application
 - Global process ID provided
 - Global signaling mechanism

- 18 admin tools
- 3 compiler families (GNU, Intel, LLVM)
- 13 development tool packages (EasyBuild, cbuild, libtool, ...)
- Lua scripting language & supporting packages
- 8 I/O libraries
 - Adios – enables defining how data is accessed
 - HDF5 – data model, library, and file format for storing and managing data
 - NetCDF – managing array-oriented scientific data
- Lustre file system
- 4 MPI packages
- 12 parallel libraries
- 14 performance tools
- Provisioning tools, resource management, runtime packages
- 6 threaded library packages

<http://openhpc.community>

HPC example: Rocks Cluster Distribution

- Employed on over 1,900 clusters (<https://app.awesome-table.com/-KIAGPC-2IYjjVG2ReJn/view>)
- Mass installation is a core part of the system
 - Mass re-installation for application-specific configurations
- Front-end central server + compute & storage nodes
- Based on CentOS Linux
- Rolls: collection of packages
 - Base roll includes: PBS (portable batch system), PVM (parallel virtual machine), MPI (message passing interface), job launchers, ...

Open-source Linux cluster distribution – supported by the National Science Foundation – rocksclusters.org

Batch Processing

Batch processing

- Non-interactive processes
 - Schedule, run eventually, collect output
- Examples:
 - MapReduce, many supercomputing tasks (circuit simulation, climate simulation, weather)
 - Graphics rendering
 - Maintain a queue of frames to be rendered
 - Have a dispatcher to remotely exec process
- In many cases – minimal or no IPC needed
- Coordinator dispatches jobs

Single-queue work distribution: Render Farms

Example – Pixar:

- 55,000 cores running RedHat Linux and Renderman (2018)
- Custom Linux software for articulating, animating/lighting (Marionette), scheduling (Ringmaster), and rendering (RenderMan)
- Toy Story
 - Each frame took between 45 minutes to 30 hours to render: 114,240 total frames
 - 117 computers running 24 hours a day
 - Toy Story 4 – 24 years later: 50-150 hours to render each frame
- Took over two years (in real time) to render Monsters University (2013)
 - Sully has over 1 million hairs – each rendered distinctly & motion animated
- Average time to render a single frame
 - Cars (2006): 8 hours
 - Cars 2 (2011): 11.5 hours
 - Disney/Pixar’s Coco – Up to 100 hours to render one frame

Batch Processing

- OpenPBS.org:
 - Portable Batch System
 - Developed by Veridian MRJ for NASA
- Commands
 - *Submit job scripts*
 - Submit interactive jobs
 - Force a job to run
 - *List jobs*
 - *Delete jobs*
 - *Hold jobs*

Load Balancing

Functions of a load balancer

- Load balancing
- Failover
- Planned outage management

Redirection

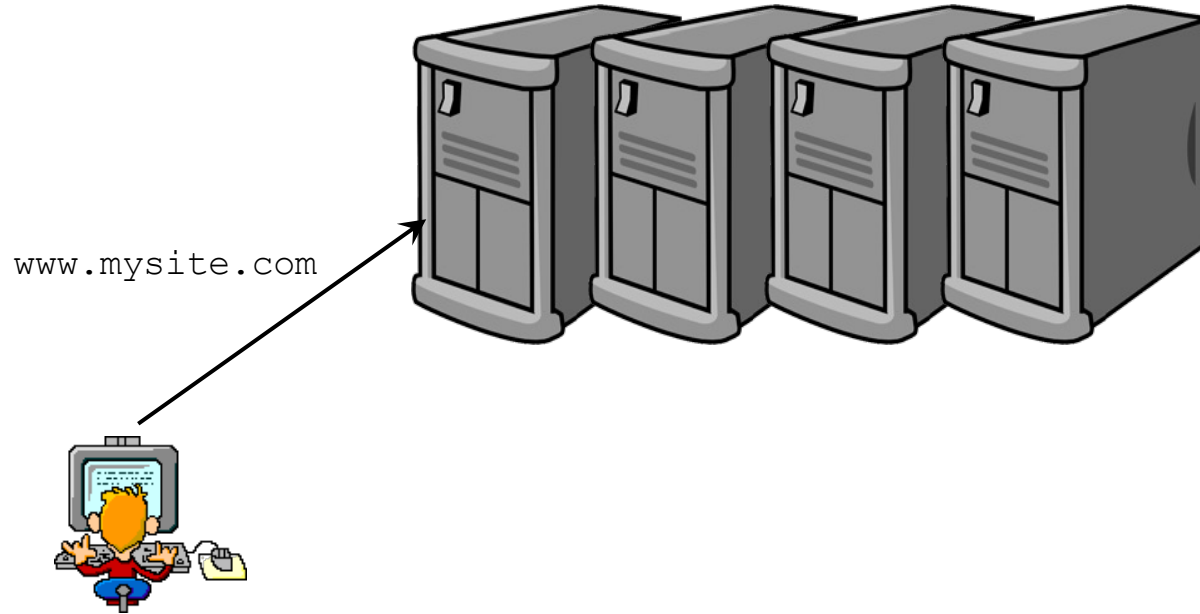
Simplest technique

HTTP REDIRECT error code

Redirection

Simplest technique

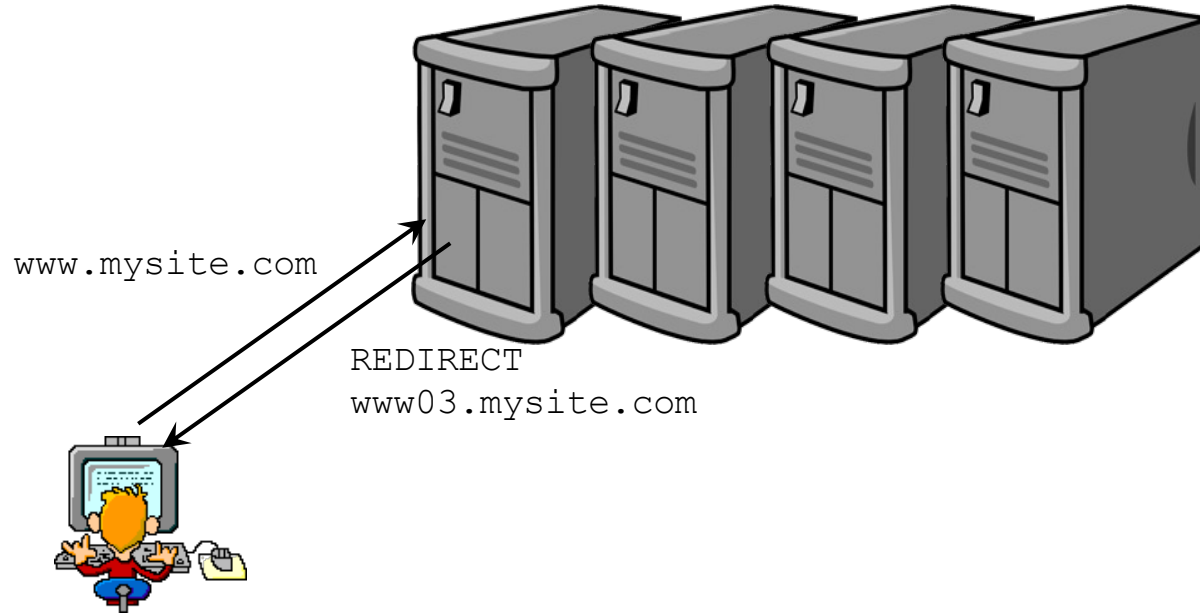
HTTP REDIRECT error code



Redirection

Simplest technique

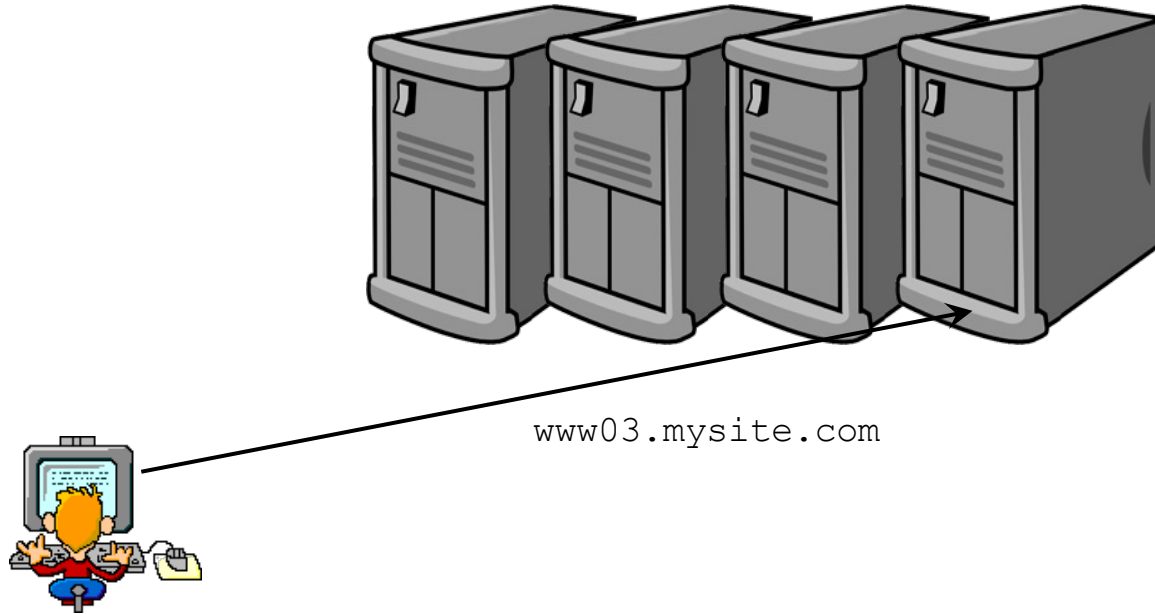
HTTP REDIRECT error code



Redirection

Simplest technique

HTTP REDIRECT error code



Redirection

- Trivial to implement
- Successive requests automatically go to the same web server
 - Important for sessions
- Visible to customer
 - Don't like the changing URL
- Bookmarks will usually tag a specific site

Load balancing router

- As routers got smarter
 - Not just simple packet forwarding
 - Most support packet filtering
 - Add load balancing to the mix

- This includes most IOS-based Cisco routers, Radware Alteon, F5 Big-IP

Load balancing router

- Assign one or more virtual addresses to physical address
 - Incoming request gets mapped to physical address
- Special assignments can be made per port
 - e.g., all FTP traffic goes to one machine
- **Balancing decisions:**
 - Pick machine with least # TCP connections
 - Factor in weights when selecting machines
 - Pick machines round-robin
 - Pick fastest connecting machine (SYN/ACK time)
- **Persistence**
 - Send all requests from one user session to the same system

DNS-based load balancing

- Round-Robin DNS
 - Respond to DNS requests with different addresses ... or a list of addresses instead of one address but the order of the list is permuted with each response
- Geographic-based DNS response
 - Multiple clusters distributed around the world
 - Balance requests among clusters
 - Favor geographic proximity
 - Examples:
 - BIND with GeoDNS patch
 - PowerDNS with Geo backend
 - Amazon Route 53



The End