Hálózatba kapcsolt erőforrás platformok és alkalmazásaik

Simon Csaba
TMIT
2019

Legal Download in EU

- http://index.hu/kultur/2013/10/30/uhd/

Blockchain alapú P2P megoldások

Zeronet

- Dynamic Web
- P2P
- Integrated BitCoin

- ZeroNet and IPFS: uncensorable auto-scaling BitTorrent powered websites
- ZeroNet is created for fast, dynamic websites, IPFS is more like a storage solution

http://bitcoinist.com/interview-developer-zeronet/

Kínai hozzáférés ?

ZeroNet
@HelloZeroNet
Követés

ZeroNet.io webpage got blocked in China: We are officially marked as threat to government censorship.

13:06 - 2017. okt. 13.

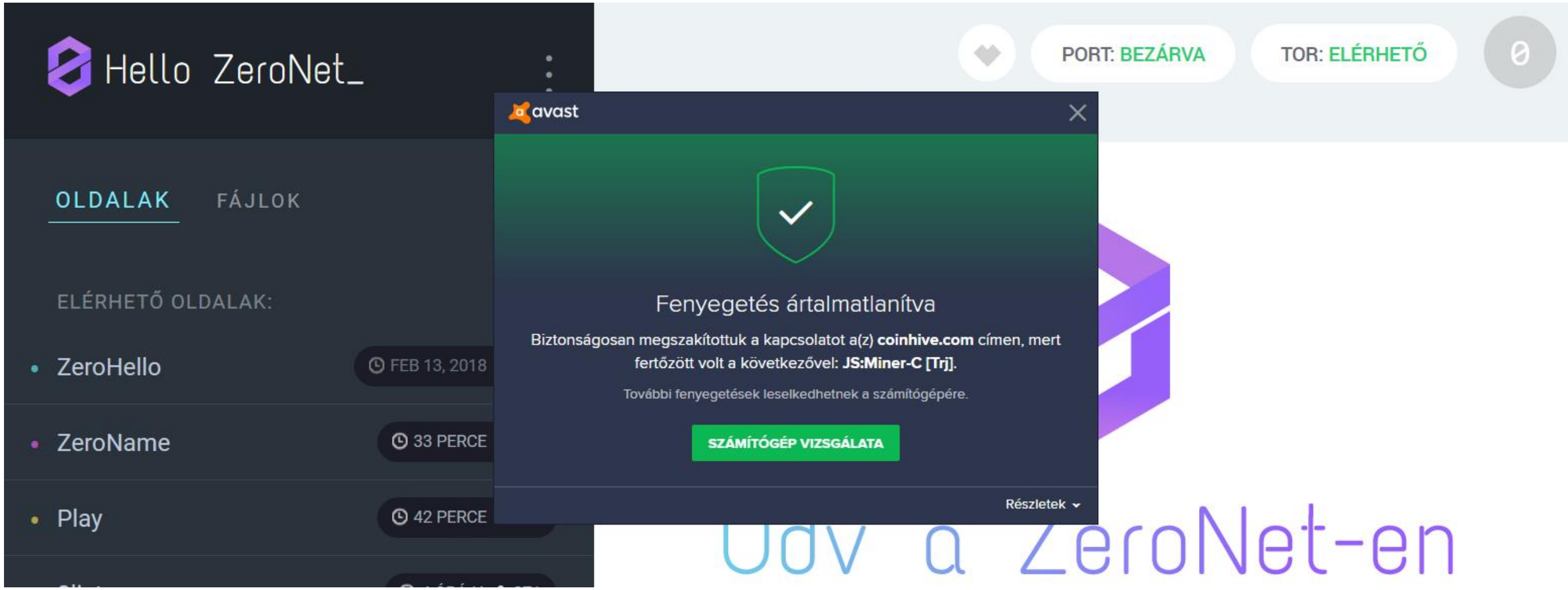
Ping: zeronet.io Go

Pinging zeronet.io, IP: 104.156.231.236 located in United States from multiple locations:

Location	ISP	Loss	Sent	Last	Avg	Best	Worst	StDev	MTR
Canada, BC, Vancouver	Telus	0%	100	26.1	25.95	25.37	26.5	0.3	show
Canada, BC, Vancouver	Shaw	0%	100	33.43	36.02	32.41	126.44	9.55	show
USA, WA, Seattle	PCCW	0%	100	18.16	18.18	17.92	18.62	0.14	show
	ayer	0%	100	17.98	18.08	17.86	18.79	0.15	show
	cane-fmt2	0%	100	1.51	1.62	1.22	11.75	1.05	show
	aNET	0%	100	13.07	13.15	12.98	15.39	0.24	show
	ayer	0%	100	38.66	38.82	38.56	39.88	0.23	show
	t	0%	100	51.46	51.7	51.07	63.37	1.67	show
	rietta/RamNode	0%	100	63.9	63.97	63.82	64.72	0.14	show
	ouse/RamNode	0%	100	78.08	78.11	77.92	78.71	0.12	show
	vIT/RamNode	0%	100	141.22	141.26	141.07	141.78	0.12	show
	lu/BuyVM	0%	100	150.35	150.59	150.13	159.24	1.27	show
	derService	0%	100	151.5	151.33	151.16	152.02	0.14	show
Italy, Milan	Prometeus	0%	100	158.14	158.39	158.09	161.52	0.52	show
Russia, Moscow	WebDC/FirstVDS	0%	100	202.2	203.77	201.74	206.64	1.86	show
Russia, Tomsk	Tomgate/Berihoster	0%	100	255.7	255.48	255.07	256.24	0.22	show
Singapore	Digital Ocean	0%	100	168.26	168.28	168.07	168.85	0.13	show
Japan, Tokyo	Vultr	0%	100	108.88	108.9	108.72	109.45	0.13	show
Australia, Sydney	Vultr	0%	100	167.14	167.01	166.75	168.34	0.22	show
China, Guizhou	China Telecom	100%	100	-	-	-	-	-	show
China, Henan	China Unicom	100%	100	-	-	-	-	-	show
China, Guangzhou	Tencent	100%	100	-	-	-	-	-	show
China, Beijing	Aliyun	100%	100	-	-	-	-	-	show
China, Shenzhen	Aliyun	100%	100	-	-	-	-	-	show
China, Anhui	China Mobile	100%	100	-	-	-	-	-	show
China, Jiangsu	China Mobile	100%	100	-	-	-	-	-	show
China, Jiangsu	China Unicom	100%	100	-	-	-	-	-	show
China, Jiangsu	China Telecom	100%	100	-	-	-	-	-	show
China, Shanghai	Aliyun	100%	100	-	-	-	-	-	show

Chart

Vírusveszély!

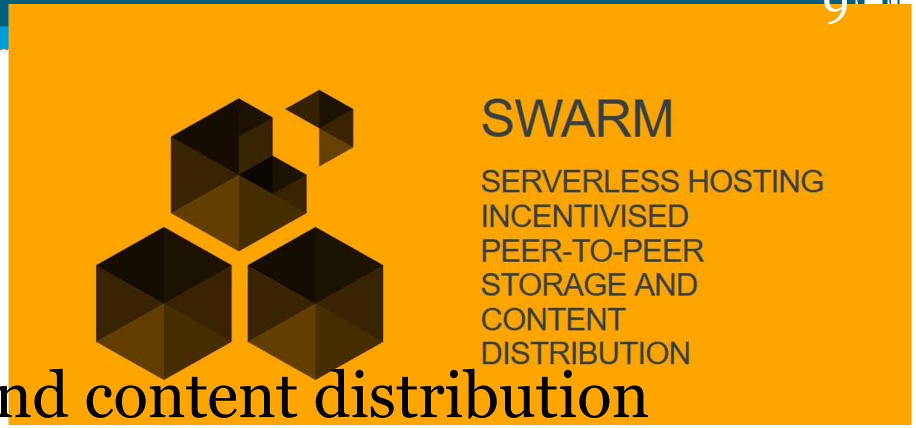


IPFS

- Interplanetary File System
  - combines the concept of linking the distributed file sharing to a file system
  - Hashing individual files
- Ethereum
  - a single shared computer that is run by the network of users and on which resources are parceled out and paid for by *Ether*
  - *finance, the internet-of-things, farm-to-table produce, electricity sourcing and pricing, and sports betting*
- Filecoin
  - incentive for storage, on top of IPFS
  - distributed electronic currency similar to Bitcoin
  - proof-of-retrievability component, which re-quires nodes to prove they store a particular file

Ethereum - https://www.cryptokitties.co/

SWARM



- Distributed **storage** platform and content distribution service
  - a native base layer service of the ethereum web 3 stack
- Features
  - DDOS-resistant, zero-downtime, fault-tolerant, censorship-resistant
  - self-sustaining due to a built-in incentive system
    - uses peer to peer accounting
    - allows trading resources for payment

http://www.epointsystem.org/~nagydani/homepage

Összefoglalás

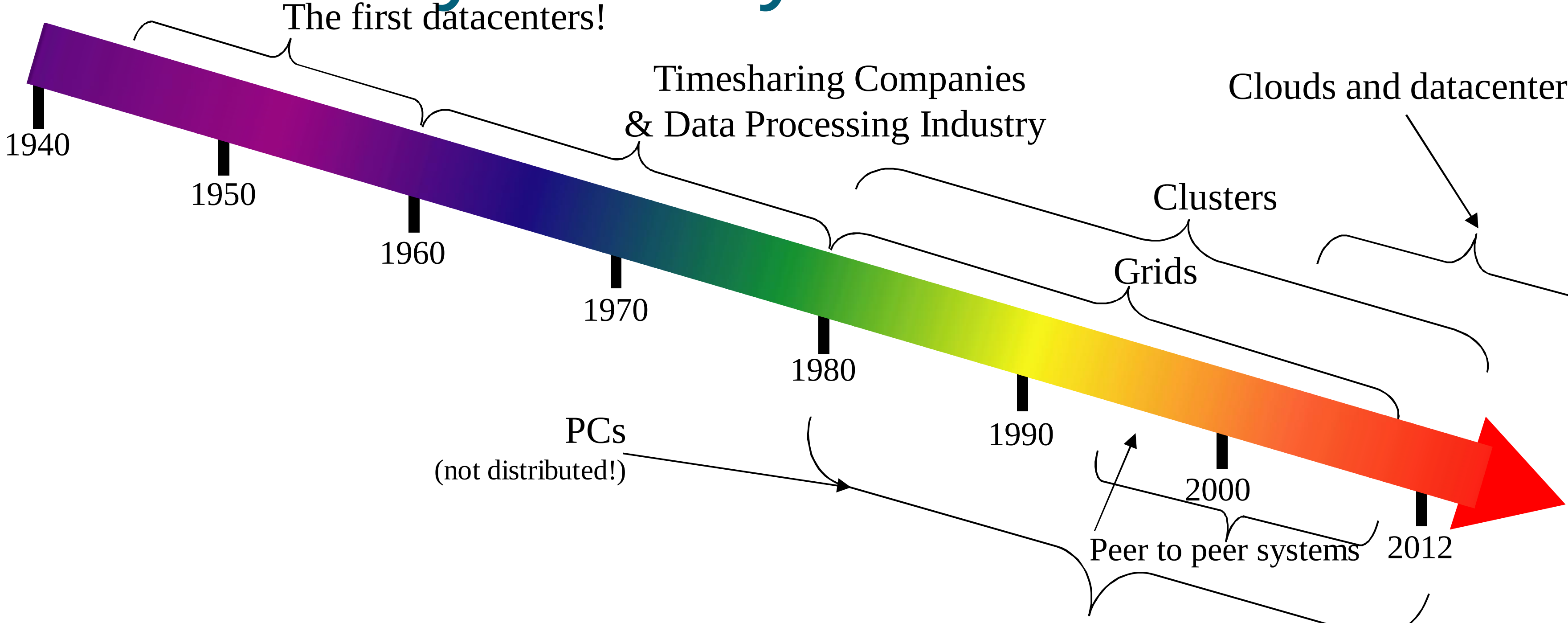
- P2P > fájlcserélés
  - Lehetőség van a résztvevők identitásának elrejtésére
  - DHT alapú
- Blockchain alapú új javaslatok (2014/15-től kezdődően)
  - Distributed storage
  - Cenzorship
  - Incentive
  - Dynamic

Klaszterek

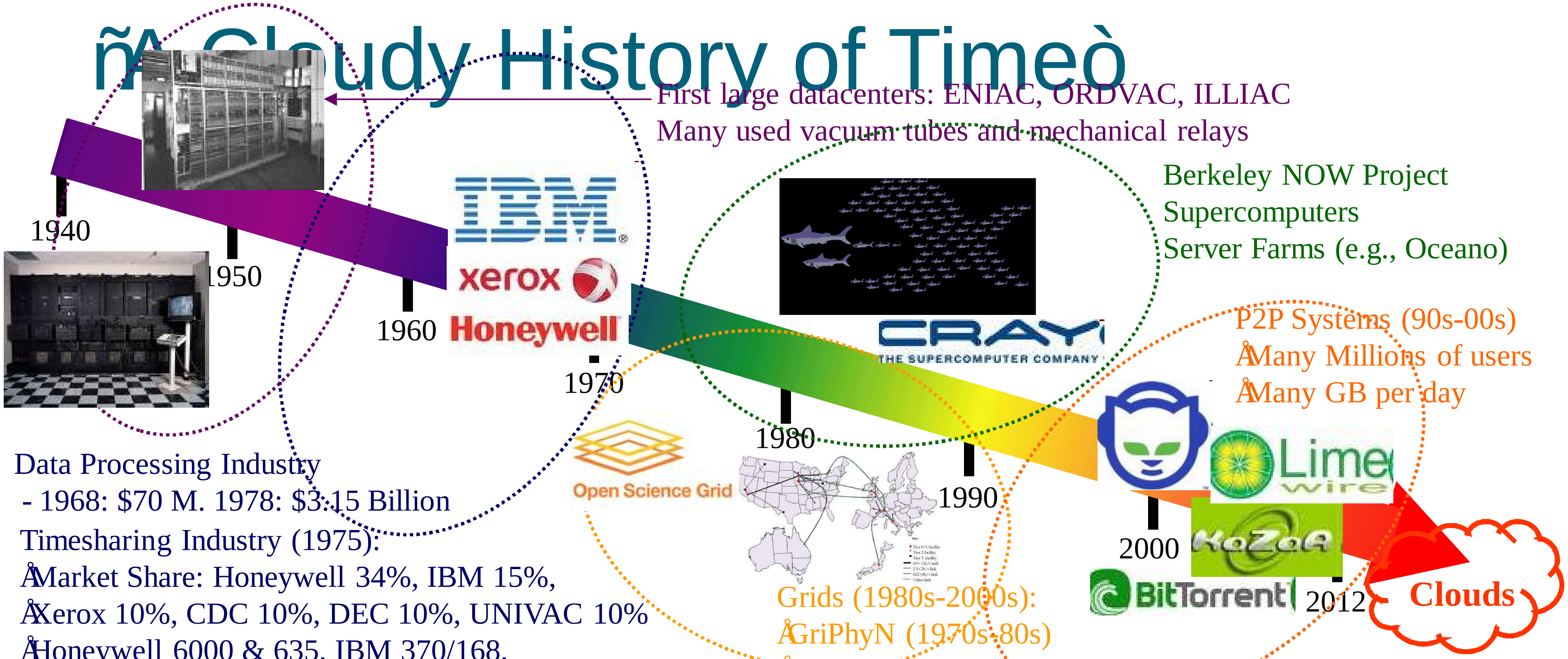
P2P Computing vs Grid Computing

- Differ in Target Communities
- Grid system deals with more complex, more powerful, more diverse and highly interconnected set of resources than P2P.
- VO

ñA Cloudy History of Timeò



ñA Cloudy History of Timeò



First large datacenters: ENIAC, ORDVAC, ILLIAC
Many used vacuum tubes and mechanical relays

Berkeley NOW Project
Supercomputers
Server Farms (e.g., Oceano)

P2P Systems (90s-00s)
ÅMany Millions of users
ÅMany GB per day

Data Processing Industry
- 1968: $70 M. 1978: $3.15 Billion

Timesharing Industry (1975):
ÅMarket Share: Honeywell 34%, IBM 15%,
ÅXerox 10%, CDC 10%, DEC 10%, UNIVAC 10%
ÅHoneywell 6000 & 635, IBM 370/168,
Xerox 940 & Sigma 9, DEC PDP-10, UNIVAC 1108

Grids (1980s-2000s):
ÅGriPhyN (1970s-80s)
ÅOpen Science Grid and Lambda Rail (2000s)
ÅGlobus & other standards (1990s-2000s)

Parelell computing

Multiprocessing

- Flynn's Taxonomy of Parallel Machines
  - How many Instruction streams?
  - How many Data streams?
- SISD: Single I Stream, Single D Stream
  - A uniprocessor
- SIMD: Single I, Multiple D Streams
  - Each "processor" works on its own data
  - But all execute the same instrs in lockstep
  - E.g. a vector processor or MMX

Flynn’s Taxonomy

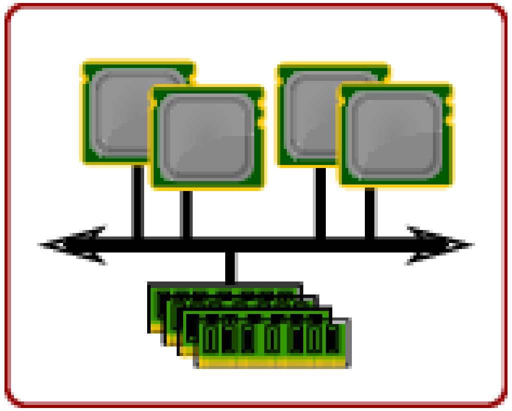
- MISD: Multiple I, Single D Stream
  - Not used much
  - Stream processors are closest to MISD
- MIMD: Multiple I, Multiple D Streams
  - Each processor executes its own instructions and operates on its own data
  - This is your typical off-the-shelf multiprocessor (made using a bunch of “normal” processors)
  - Includes multi-core processors

Multiprocessors

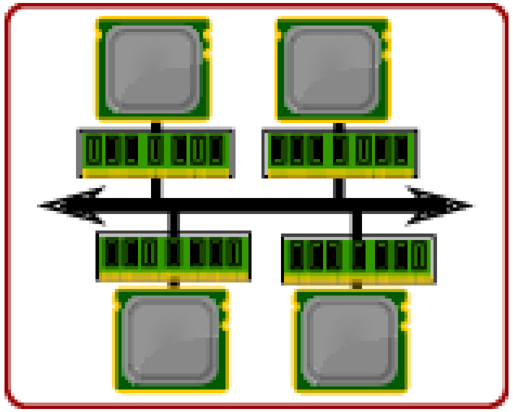
- Why do we need multiprocessors?
  - Uniprocessor speed keeps improving
  - But there are things that need even more speed
    - Wait for a few years for Moore's law to catch up?
    - Or use multiple processors and do it now?
- Multiprocessor software problem
  - Most code is sequential (for uniprocessors)
    - MUCH easier to write and debug
  - Correct parallel code very, very difficult to write
    - ***Efficient*** and correct is even harder
    - Debugging even more difficult (Heisenbugs)

MIMD Multiprocessors

Centralized Shared Memory

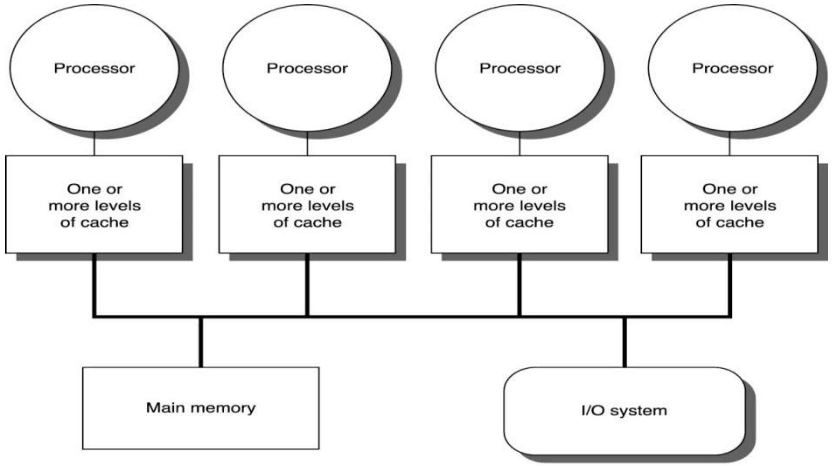


Distributed Memory

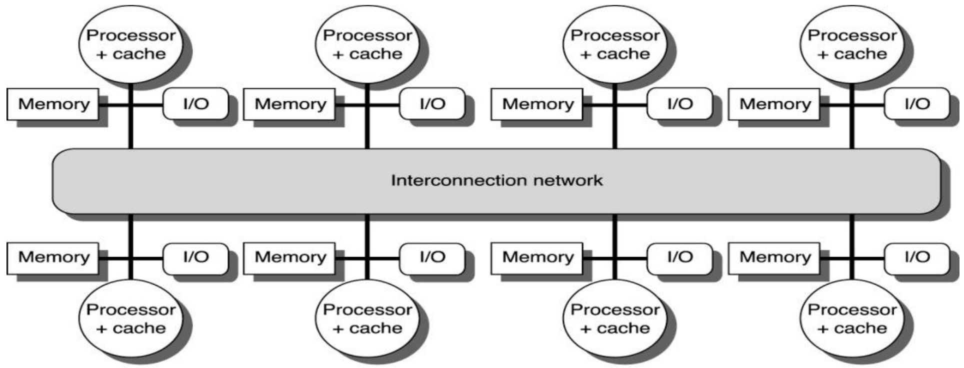


MIMD Multiprocessors

Centralized Shared Memory



Distributed Memory



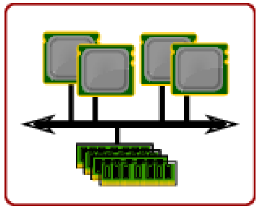
Centralized-Memory Machines

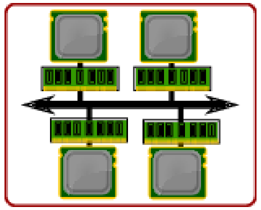
- Also “Symmetric Multiprocessors” (SMP)
- “Uniform Memory Access” (UMA)
  - All memory locations have similar latencies
  - Data sharing through memory reads/writes
  - P1 can write data to a physical address A, P2 can then read physical address A to get that data
- Problem: Memory Contention
  - All processor share the one memory
  - Memory bandwidth becomes bottleneck
  - Used only for smaller machines
    - Most often 2,4, or 8 processors

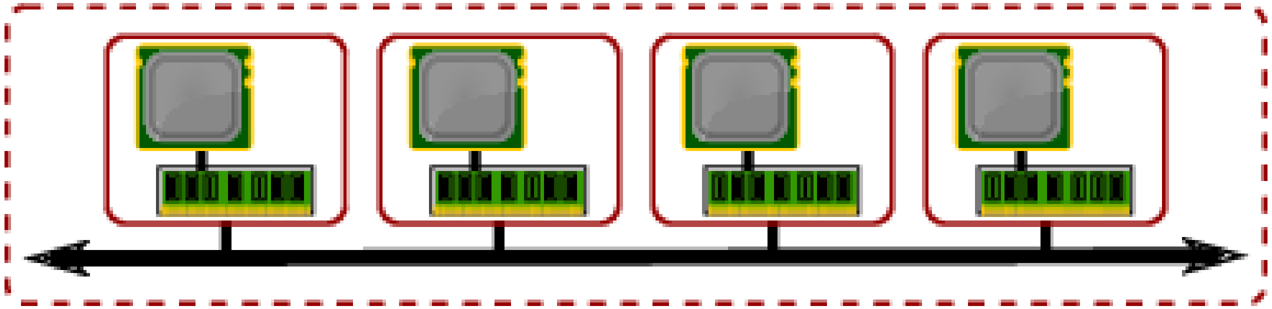
Distributed-Memory Machines

- Two kinds
  - Distributed Shared-Memory (DSM)
    - All processors can address all memory locations
    - Data sharing like in SMP
    - Also called **NUMA** (non-uniform memory access)
    - Latencies of different memory locations can differ (local access faster than remote access)
  - Message-Passing
    - A processor can directly address only local memory
    - To communicate with other processors, must explicitly send/receive messages
    - Also called multicomputers or clusters
- Most accesses local, so less memory contention (can scale to well over 1000 processors)

Message-Passing Machines





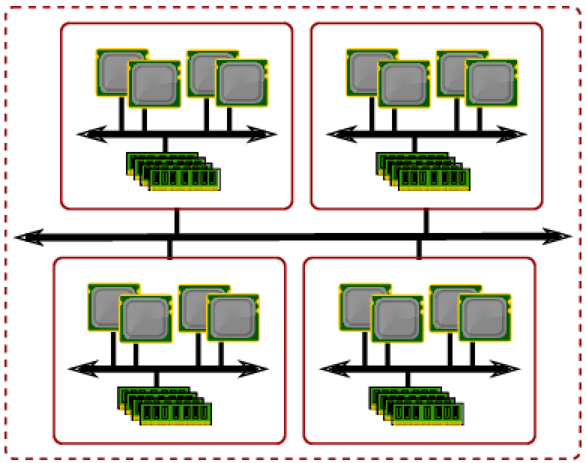


Message-Passing Machines

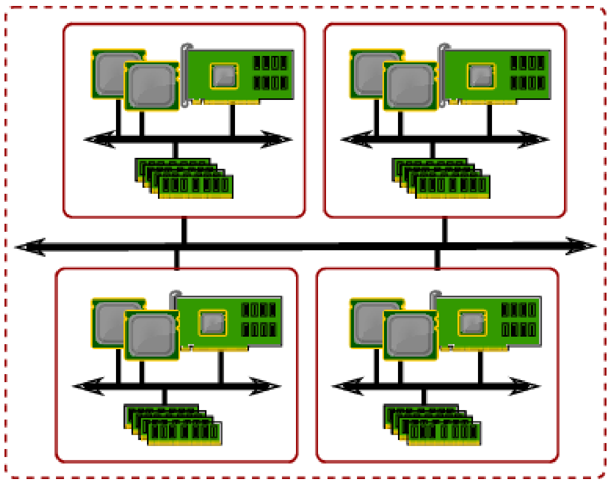
- A cluster of computers
  - Each with its own processor and memory
  - An interconnect to pass messages between them
  - Producer-Consumer Scenario:
    - P1 produces data D, uses a SEND to send it to P2
    - The network routes the message to P2
    - P2 then calls a RECEIVE to get the message
  - Two types of send primitives
    - Synchronous: P1 stops until P2 confirms receipt of message
    - Asynchronous: P1 sends its message and continues
  - Standard libraries for message passing:
    Most common is MPI – Message Passing Interface

Hybrid architectures

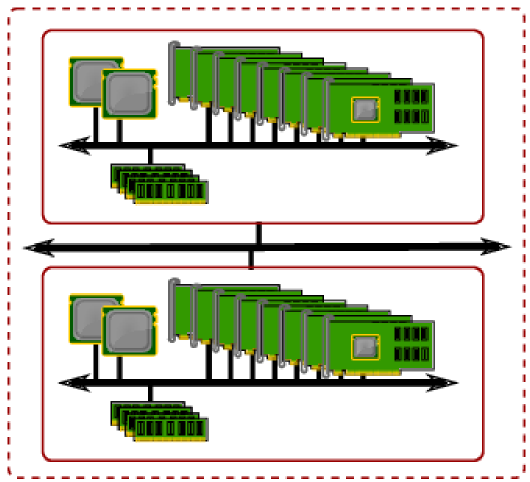
Fat cluster



GPU-accelerated



GPU cluster



Communication Performance

- Metrics for Communication Performance
  - Communication Bandwidth
  - Communication Latency
    - Sender overhead + transfer time + receiver overhead
  - Communication latency hiding
- Characterizing Applications
  - Communication to Computation Ratio
    - Work done vs. bytes sent over network
    - Example: 146 bytes per 1000 instructions

Message Passing Interface

Message Passing Pros and Cons

- Pros
  - Simpler and cheaper hardware
  - Explicit communication makes programmers aware of costly (communication) operations
- Cons
  - Explicit communication is painful to program
  - Requires manual optimization
    - If you want a variable to be local and accessible via LD/ST, you must declare it as such
    - If other processes need to read or write this variable, you must explicitly code the needed sends and receives to do this

Message Passing: A Program

- Calculating the sum of array elements

```
#define ASIZE 1024
#define NUMPROC 4
double myArray[ASIZE/NUMPROC];
double mySum=0;
for(int i=0;i<ASIZE/NUMPROC;i++)
  mySum+=myArray[i];
if(myPID=0){
  for(int p=1;p<NUMPROC;p++){
    int pSum;
    recv(p,pSum);
    mySum+=pSum;
  }
  printf("Sum: %lf\n",mySum);
}else
  send(0,mySum);
```

Must manually split the array

ñMasterò processor adds up partial sums and prints the result

ñSlaveò processors send their partial results to master

MPI programming example

- https://hpcc.usc.edu/support/documentation/examples-of-mpi-programs

Shared Memory Pros and Cons

- Pros
  - Communication happens automatically
  - More natural way of programming
    - Easier to write correct programs and gradually optimize them
  - No need to manually distribute data
    (but can help if you do)
- Cons
  - Needs more hardware support
  - Easy to write correct, but inefficient programs
    (remote accesses look the same as local ones)

High-Performance Computing / Introduction

Source: James R. Knight/Yale Center for Genome Analysis

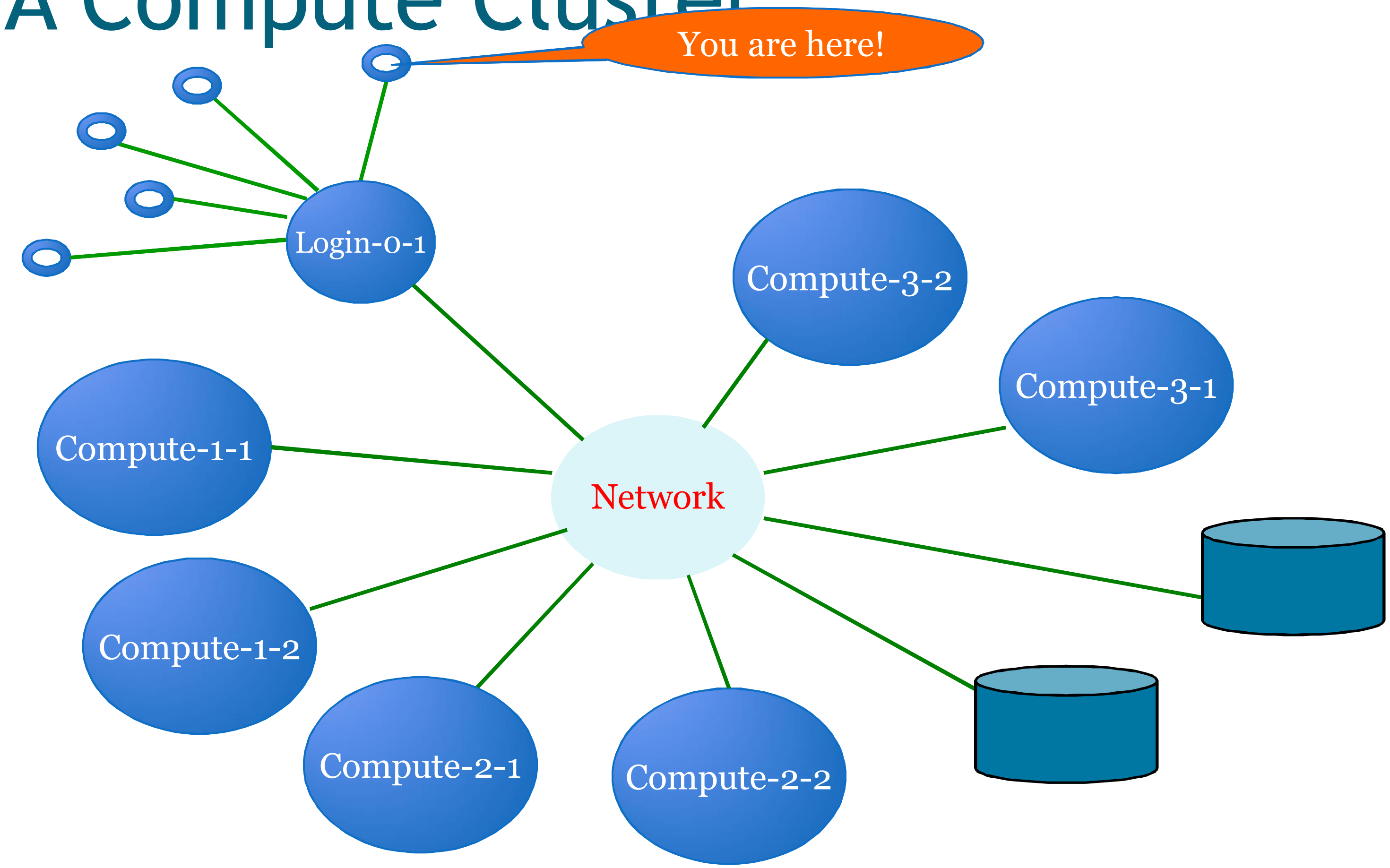
1950's – The Beginning...

2016 – Looking very similar...

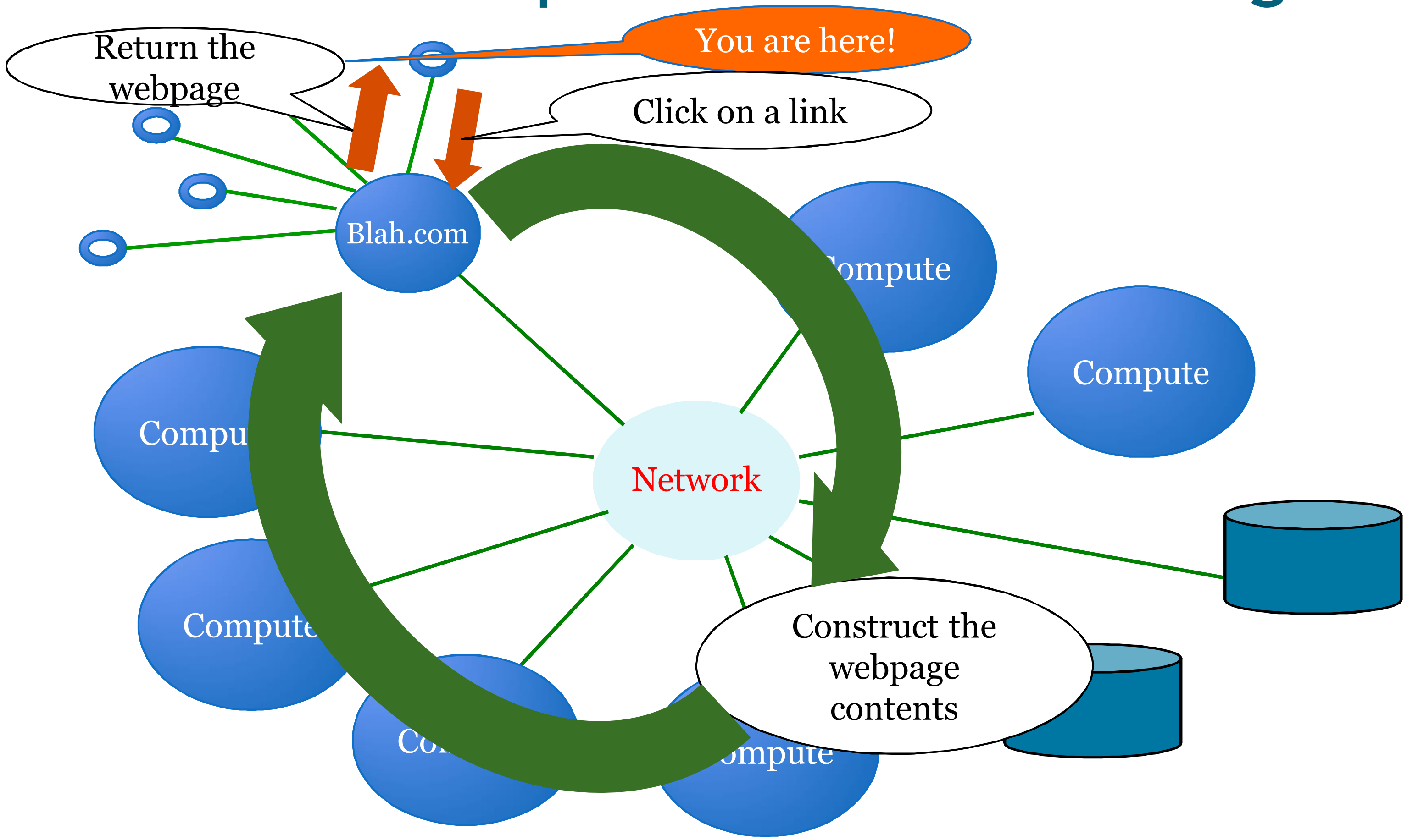
…but there are differences

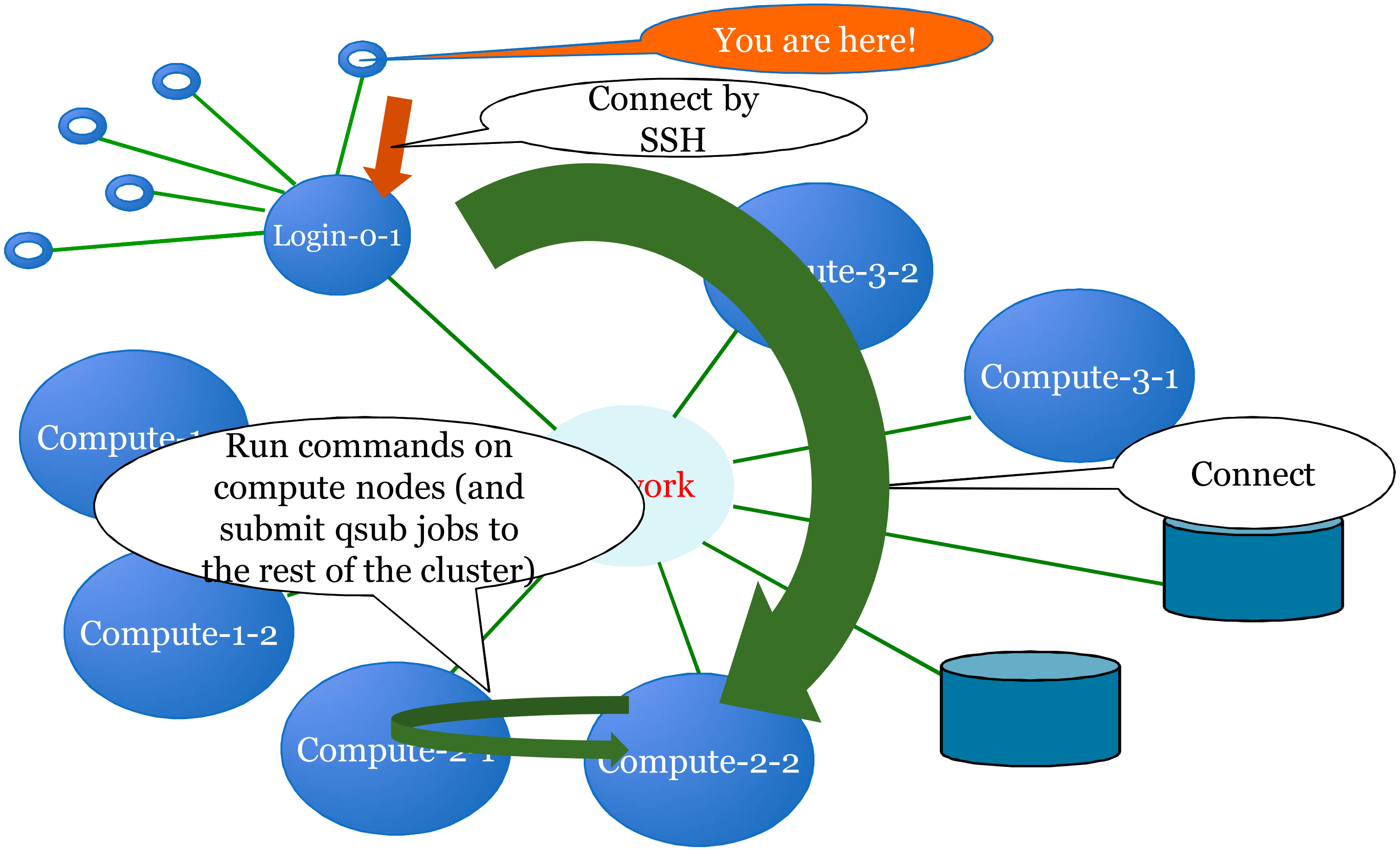
- Not a single computer but thousands of them, called a **cluster**
  - Hundreds of physical "computers", called **nodes**
  - Each with 4-64 CPU's, called **cores**
- Nobody works in the server rooms anymore
  - IT is there to fix what breaks, not to run computations (or help you run computations)
  - Everything is done by remote connections
- Computation is performed by submitting **jobs** for running
  - This actually hasn't changed…but how you run jobs has…

A Compute Cluster



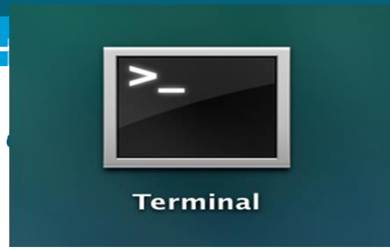
You Use a Compute Cluster! Surfing the Web



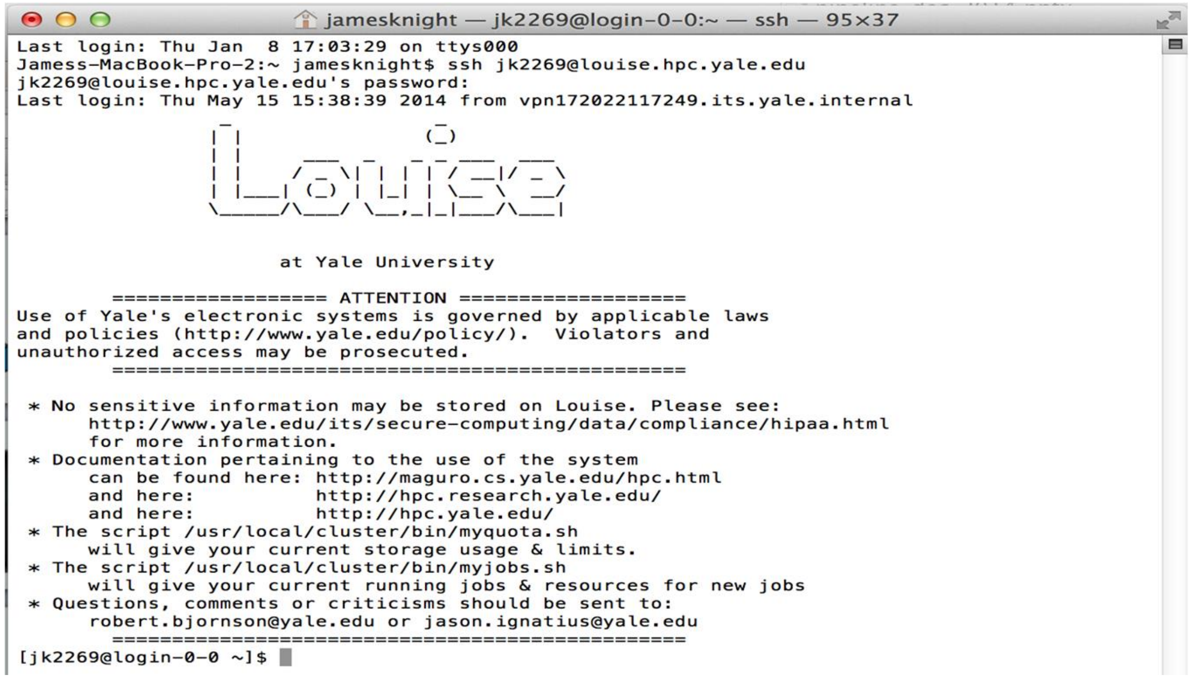

You are here!
Connect by SSH
Login-0-1
Compute-3-2
Compute-3-1
Connect
Run commands on compute nodes (and submit qsub jobs to the rest of the cluster)
Compute-1-2
Compute-2-1
Compute-2-2

1970’s – Terminals, In the Beginning...

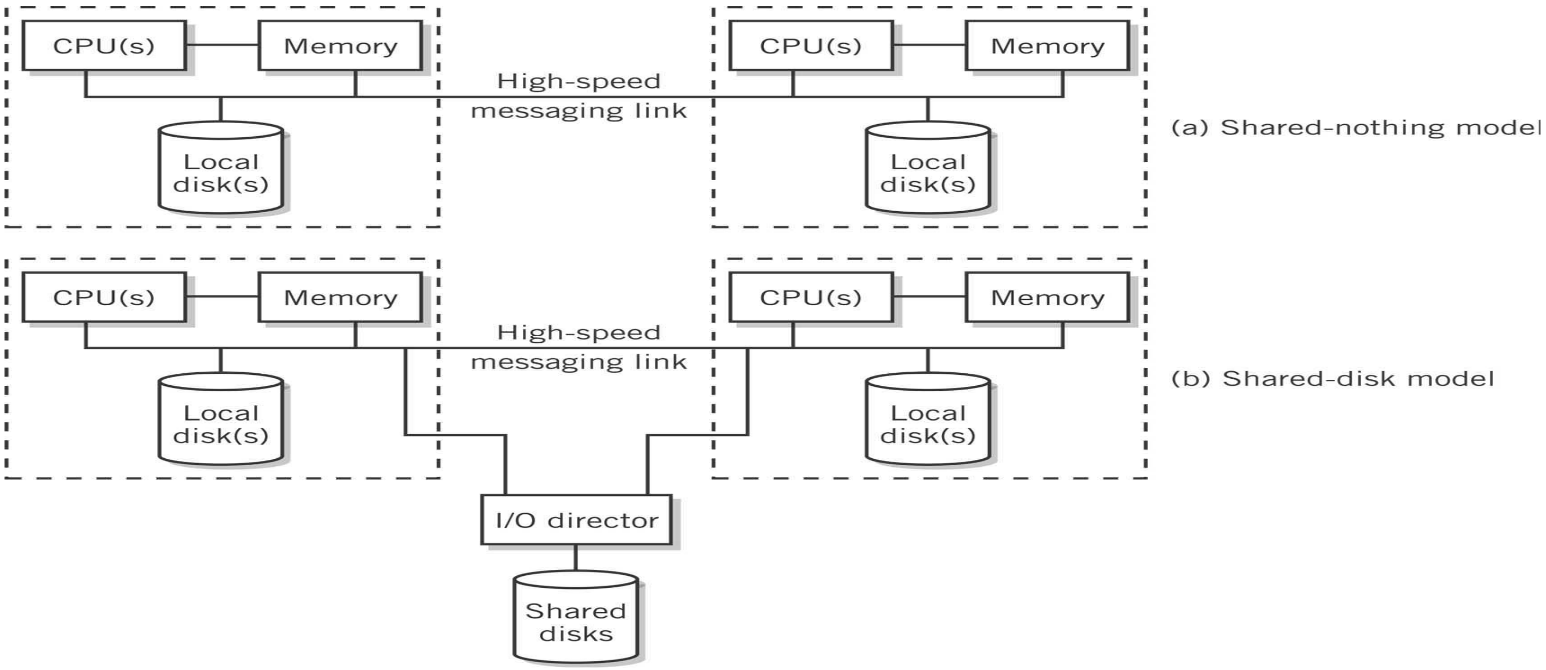
2016 – Pretty much the same…



- Terminal app on Mac
- Look in the "Other" folder in Launchpad



Cluster Models



Beowulf Clusters

- Simple and highly configurable
- Low cost
- Networked
  - Computers connected to one another by a private Ethernet network
  - Connection to an external network is through a single gateway computer
- Configuration
  - COTS – Commodity-off-the-shelf components such as inexpensive computers
  - Blade components – computers mounted on a motherboard that are plugged into connectors on a rack
  - Either shared-disk or shared-nothing model

Blade and Rack of Beowulf Cluster

Cluster computing concept

Cluster Computing - Research Projects

- Beowulf (CalTech and NASA) - USA
- CCS (Computing Centre Software) - Paderborn, Germany
- Condor - Wisconsin State University, USA
- DQS (Distributed Queuing System) - Florida State University, US.
- EASY - Argonne National Lab, USA
- HPVM -(High Performance Virtual Machine),UIUC&now UCSB,US
- *far* - University of Liverpool, UK
- Gardens - Queensland University of Technology, Australia
- MOSIX - Hebrew University of Jerusalem, Israel
- MPI (MPI Forum, MPICH is one of the popular implementations)
- NOW (Network of Workstations) - Berkeley, USA
- NIMROD - Monash University, Australia
- NetSolve - University of Tennessee, USA
- PBS (Portable Batch System) - NASA Ames and LLNL, USA
- PVM - Oak Ridge National Lab./UTK/Emory, USA

Cluster Computing - Commercial Software

- Codine (Computing in Distributed Network Environment) - GENIAS GmbH, Germany
- LoadLeveler - IBM Corp., USA
- LSF (Load Sharing Facility) - Platform Computing, Canada
- NQE (Network Queuing Environment) - Craysoft Corp., USA
- OpenFrame - Centre for Development of Advanced Computing, India
- RWPC (Real World Computing Partnership), Japan
- Unixware (SCO-Santa Cruz Operations,), USA
- Solaris-MC (Sun Microsystems), USA
- ClusterTools (A number for free HPC clusters tools from Sun)
- A number of commercial vendors worldwide are offering clustering solutions including IBM, Compaq, Microsoft, a number of startups like TurboLinux, HPTI, Scali, BlackStone.....)

Motivation for using Clusters

- Surveys show utilisation of CPU cycles of desktop workstations is typically <10%.
- Performance of workstations and PCs is rapidly improving
- As performance grows, percent utilisation will decrease even further!
- Organisations are reluctant to buy large supercomputers, due to the large expense and short useful life span.

Motivation for using Clusters

- The development tools for workstations are more mature than the contrasting proprietary solutions for parallel computers - mainly due to the non-standard nature of many parallel systems.
- Workstation clusters are a cheap and readily available alternative to specialised High Performance Computing (HPC) platforms.
- Use of clusters of workstations as a distributed compute resource is very cost effective - incremental growth of system!!!

Cycle Stealing

- Usually a workstation will be *owned* by an individual, group, department, or organisation - they are dedicated to the exclusive use by the *owners*.
- This brings problems when attempting to form a cluster of workstations for running distributed applications.

Cycle Stealing

- Typically, there are three types of owners, who use their workstations mostly for:
  1. Sending and receiving email and preparing documents.
  2. Software development - edit, compile, debug and test cycle.
  3. Running compute-intensive applications.

Cycle Stealing

- Cluster computing aims to steal spare cycles from (1) and (2) to provide resources for (3).
- However, this requires overcoming the *ownership hurdle* - people are very protective of *their* workstations.
- Usually requires organisational mandate that computers are to be used in this way.
- Stealing cycles outside standard work hours (e.g. overnight) is easy, stealing idle cycles during work hours without impacting interactive use (both CPU and memory) is much harder.

Type of Clusters

- HA
- Load distribution

High-Performance Computing / Introduction

Source: James R. Knight/Yale Center for Genome Analysis

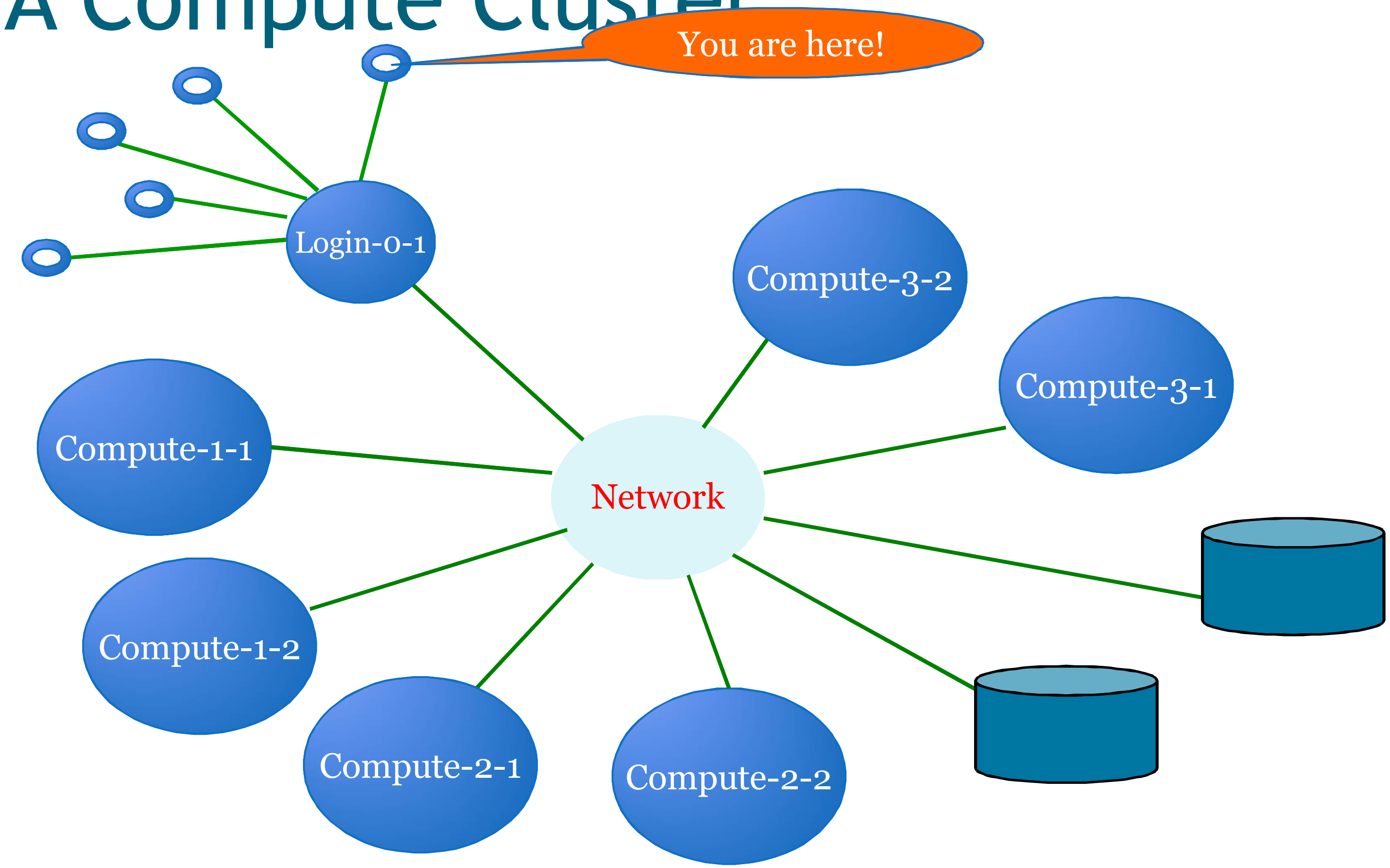
1950’s – The Beginning...

2016 – Looking very similar…

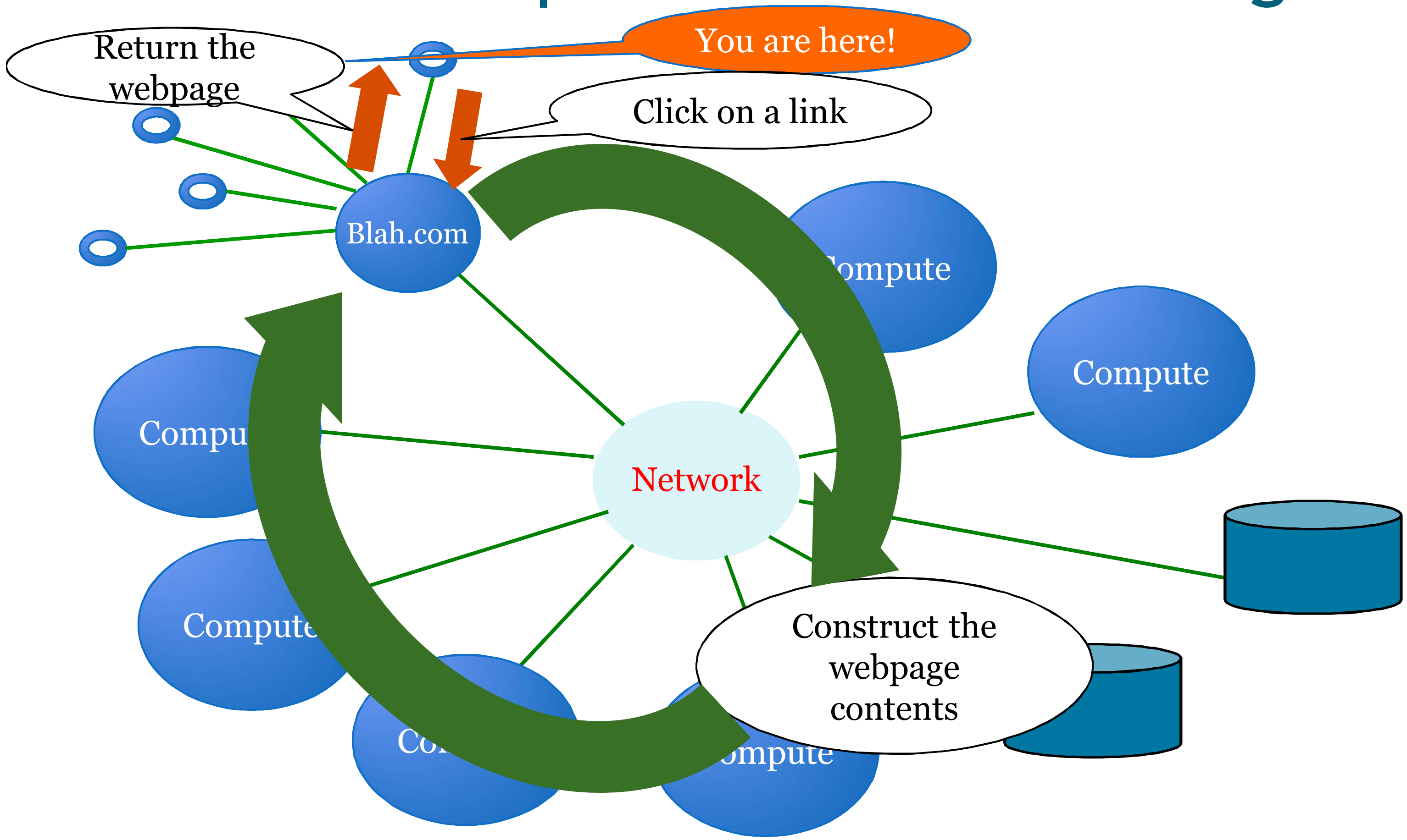
…but there are differences

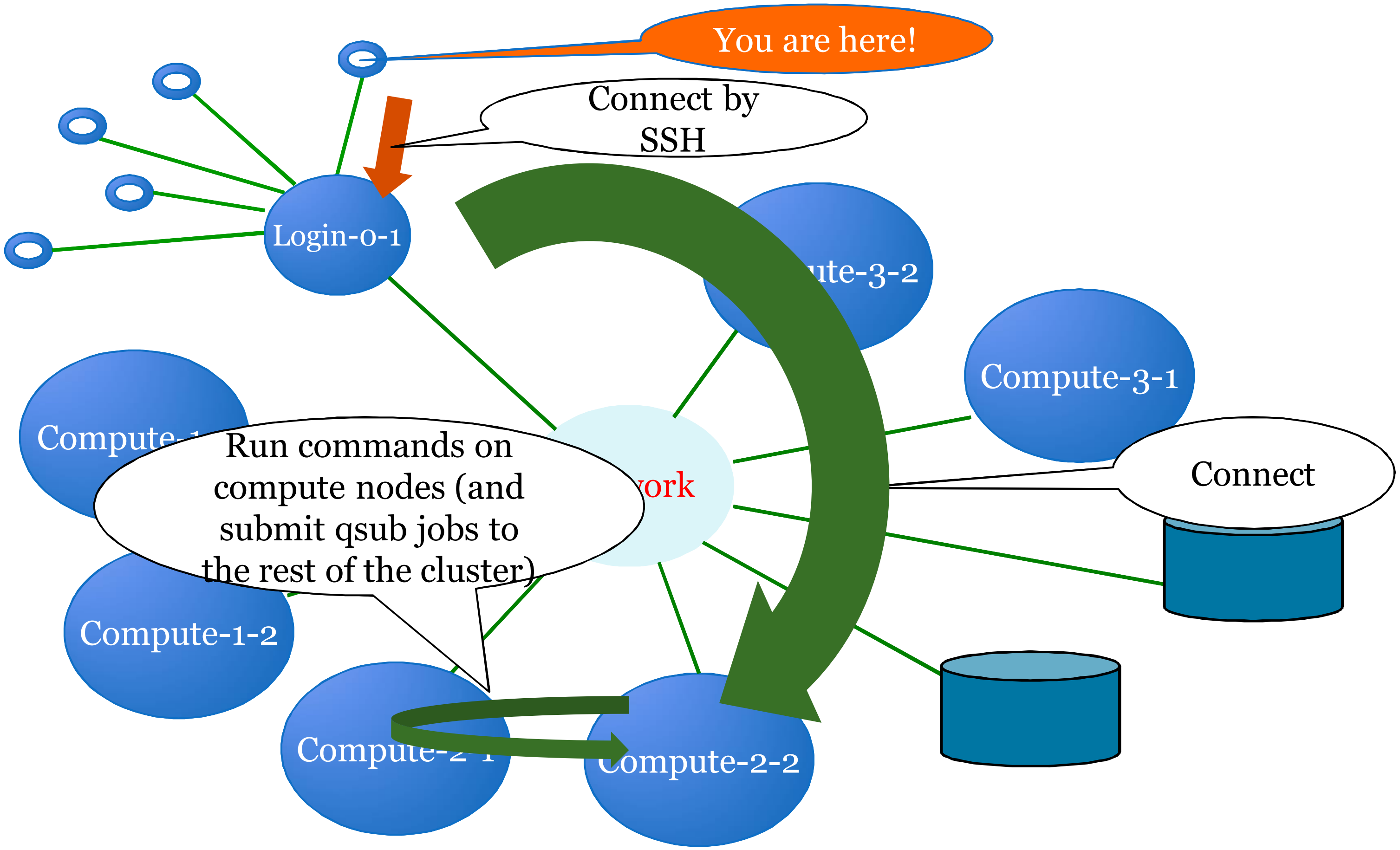
- Not a single computer but thousands of them, called a **<u>cluster</u>**
  - Hundreds of physical "computers", called **<u>nodes</u>**
  - Each with 4-64 CPU's, called **<u>cores</u>**
- Nobody works in the server rooms anymore
  - IT is there to fix what breaks, not to run computations (or help you run computations)
  - Everything is done by remote connections
- Computation is performed by submitting **<u>jobs</u>** for running
  - This actually hasn't changed…but how you run jobs has…

A Compute Cluster



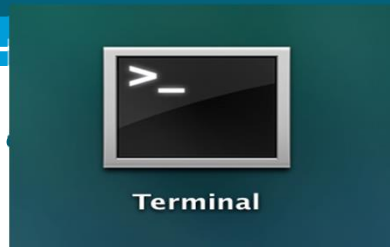
You Use a Compute Cluster! Surfing the Web



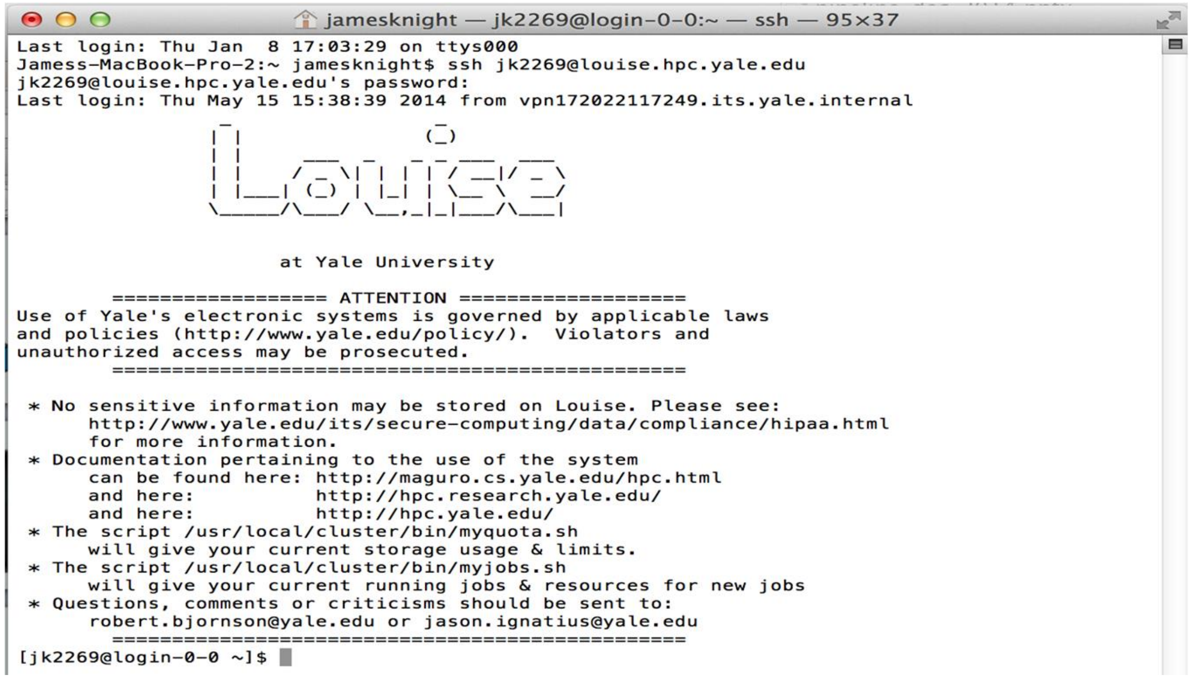

You are here!
Connect by SSH
Login-0-1
ute-3-2
Compute-3-1
Compute-1
Run commands on compute nodes (and submit qsub jobs to the rest of the cluster)
ork
Connect
Compute-1-2
Compute-2-1
Compute-2-2

1970's – Terminals, In the Beginning...

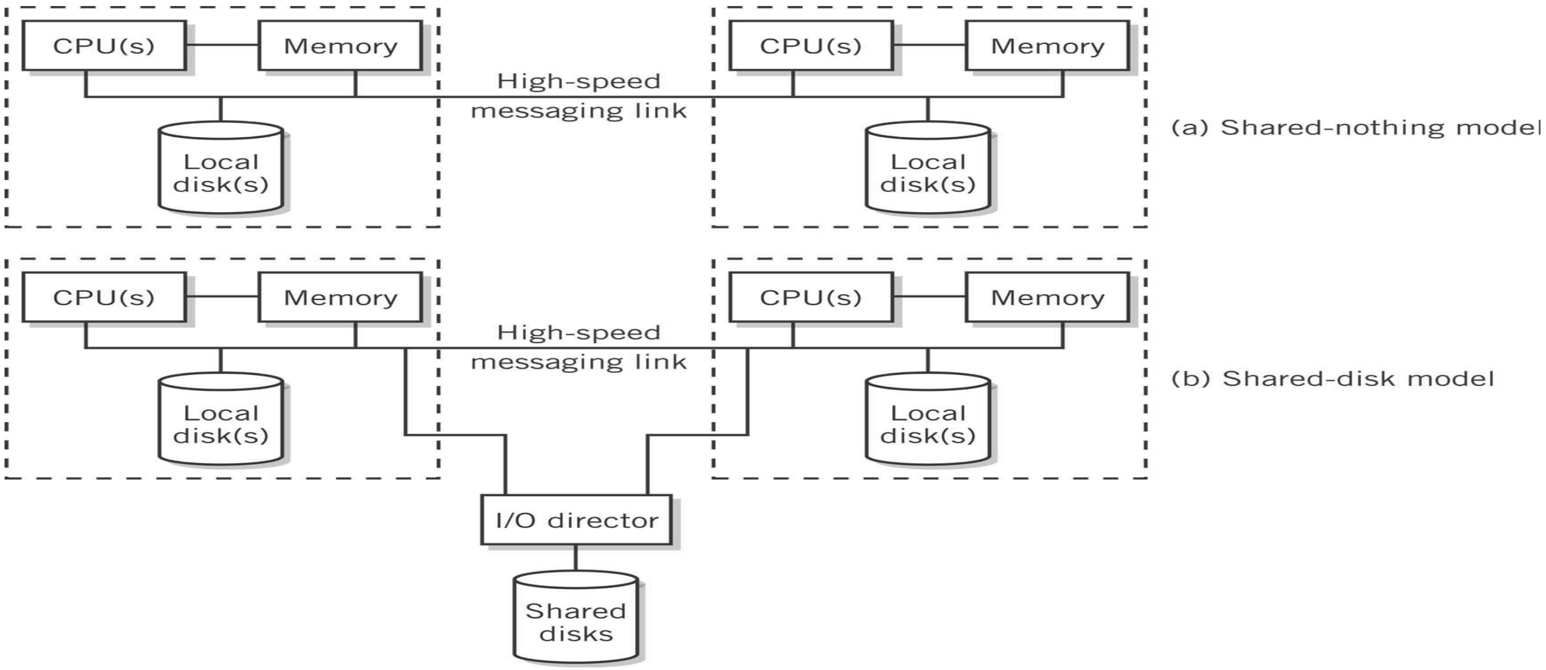
2016 – Pretty much the same.



- Terminal app on Mac
- Look in the “Other” folder in Launchpad



Cluster Models



Beowulf Clusters

- Simple and highly configurable
- Low cost
- Networked
  - Computers connected to one another by a private Ethernet network
  - Connection to an external network is through a single gateway computer
- Configuration
  - COTS – Commodity-off-the-shelf components such as inexpensive computers
  - Blade components – computers mounted on a motherboard that are plugged into connectors on a rack
  - Either shared-disk or shared-nothing model

Blade and Rack of Beowulf Cluster

Cluster computing concept

Cluster Computing - Research Projects

- Beowulf (CalTech and NASA) - USA
- CCS (Computing Centre Software) - Paderborn, Germany
- Condor - Wisconsin State University, USA
- DQS (Distributed Queuing System) - Florida State University, US.
- EASY - Argonne National Lab, USA
- HPVM -(High Performance Virtual Machine),UIUC&now UCSB,US
- *far* - University of Liverpool, UK
- Gardens - Queensland University of Technology, Australia
- MOSIX - Hebrew University of Jerusalem, Israel
- MPI (MPI Forum, MPICH is one of the popular implementations)
- NOW (Network of Workstations) - Berkeley, USA
- NIMROD - Monash University, Australia
- NetSolve - University of Tennessee, USA
- PBS (Portable Batch System) - NASA Ames and LLNL, USA
- PVM - Oak Ridge National Lab./UTK/Emory, USA

Cluster Computing - Commercial Software

- Codine (Computing in Distributed Network Environment) - GENIAS GmbH, Germany
- LoadLeveler - IBM Corp., USA
- LSF (Load Sharing Facility) - Platform Computing, Canada
- NQE (Network Queuing Environment) - Craysoft Corp., USA
- OpenFrame - Centre for Development of Advanced Computing, India
- RWPC (Real World Computing Partnership), Japan
- Unixware (SCO-Santa Cruz Operations,), USA
- Solaris-MC (Sun Microsystems), USA
- ClusterTools (A number for free HPC clusters tools from Sun)
- A number of commercial vendors worldwide are offering clustering solutions including IBM, Compaq, Microsoft, a number of startups like TurboLinux, HPTI, Scali, BlackStone.....)

Motivation for using Clusters

- Surveys show utilisation of CPU cycles of desktop workstations is typically <10%.
- Performance of workstations and PCs is rapidly improving
- As performance grows, percent utilisation will decrease even further!
- Organisations are reluctant to buy large supercomputers, due to the large expense and short useful life span.

Motivation for using Clusters

- The development tools for workstations are more mature than the contrasting proprietary solutions for parallel computers - mainly due to the non-standard nature of many parallel systems.
- Workstation clusters are a cheap and readily available alternative to specialised High Performance Computing (HPC) platforms.
- Use of clusters of workstations as a distributed compute resource is very cost effective - incremental growth of system!!!

Cycle Stealing

- Usually a workstation will be *owned* by an individual, group, department, or organisation - they are dedicated to the exclusive use by the *owners*.
- This brings problems when attempting to form a cluster of workstations for running distributed applications.

Cycle Stealing

- Typically, there are three types of owners, who use their workstations mostly for:
  1. Sending and receiving email and preparing documents.
  2. Software development - edit, compile, debug and test cycle.
  3. Running compute-intensive applications.

Cycle Stealing

- Cluster computing aims to steal spare cycles from (1) and (2) to provide resources for (3).
- However, this requires overcoming the *ownership hurdle* - people are very protective of *their* workstations.
- Usually requires organisational mandate that computers are to be used in this way.
- Stealing cycles outside standard work hours (e.g. overnight) is easy, stealing idle cycles during work hours without impacting interactive use (both CPU and memory) is much harder.

Type of Clusters

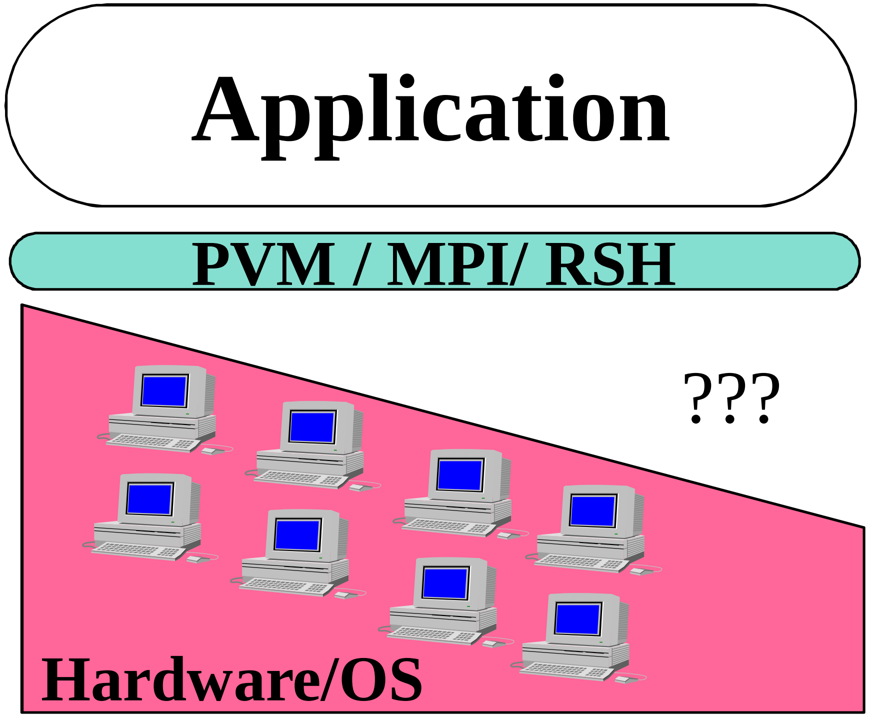
- HA
- Load distribution

P2P Computing vs Cluster/Grid Computing

- Differ in Target Communities
- Grid system deals with more complex, more powerful, more diverse and highly interconnected set of resources than P2P.

Cluster Work Schedulers

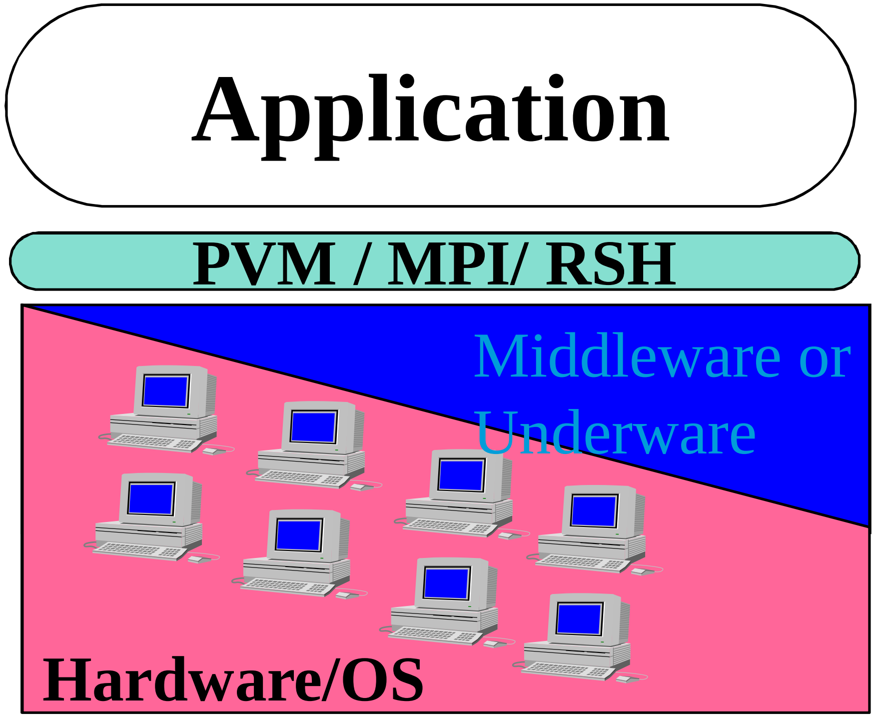
A typical Cluster Computing Environment



CC should support

- Multi-user, time-sharing environments
- Nodes with different CPU speeds and memory sizes (heterogeneous configuration)
- Many processes, with unpredictable requirements
- Unlike SMP: insufficient “bonds” between nodes
  - Each computer operates independently
  - Inefficient utilization of resources

The missing link is provide by cluster middleware/underware



SSI Clusters--SMP services on a CC

ñ*Pool Togetherò* the ñ*Cluster-Wideò* resources

- Adaptive resource usage for better performance
- Ease of use - almost like SMP
- Scalable configurations - by decentralized control

Result: *HPC/HAC at PC/Workstation prices*

What is Cluster Middleware ?

- An interface between between use applications and cluster hardware and OS platform.
- Middleware packages support each other at the management, programming, and implementation levels.
- Middleware Layers:
  - SSI Layer
  - Availability Layer: It enables the cluster services of
    - Checkpointing, Automatic Failover, recovery from failure,
    - fault-tolerant operating among all cluster nodes.

Middleware Design Goals

- Complete Transparency (Manageability)
  - Lets the see a single cluster system..
    - Single entry point, ftp, telnet, software loading...
- Scalable Performance
  - Easy growth of cluster
    - no change of API & automatic load distribution.
- Enhanced Availability
  - Automatic Recovery from failures
    - Employ checkpointing & fault tolerant technologies
  - Handle consistency of data when replicated..

Work schedulers - requirements

- Interactive or batch

- Stable
- Robust
- Efficient resource management
- Lightweigth
- Fair
- Avoids starvation

- SGE - Sun Grid Engine (Oracle Grid Engine, Open Grid Scheduler)
- SLURM (Simple Linux Utility for Resource Management)
- MOAB + Torque
- HTCondor
- ...

Resource Manager (RM)

- While other systems may have more strict interpretations of a resource manager and its responsibilities, Moab's *multi-resource manager* support allows a much more liberal interpretation.
  - In essence, any object which can provide environmental information and environmental control can be utilized as a resource manager.
- Moab is able to aggregate information from multiple unrelated sources into a larger more complete *world view* of the cluster which includes all the information and control found within a standard resource manager such as TORQUE including:
  - Node
  - Job
  - Queue management services.