Grid Computing Concepts and Tools

COMPUTE SCHOOL COURSE/WORKSHOP

Outline

Outline of the course/workshop

- 8 lessons:
  - ~45 min lecture + ~45 min hands-on exercises
- Home assignments
  - Extended hands-on work
- Final project
  - Porting of a familiar task to Grid
  - Students are expected to demonstrate understanding of basics of Grid computing and ability to work in a Grid environment

Lessons

1. Introduction: from traditional computing to Grid
2. Security and certificates
3. Delegation and Virtual Organisations
4. Computing services
5. Scientific data management
6. Information and monitoring
7. Scheduling, clients
8. Runtime environment
9. Project summaries

Introduction

BASIC CONCEPTS

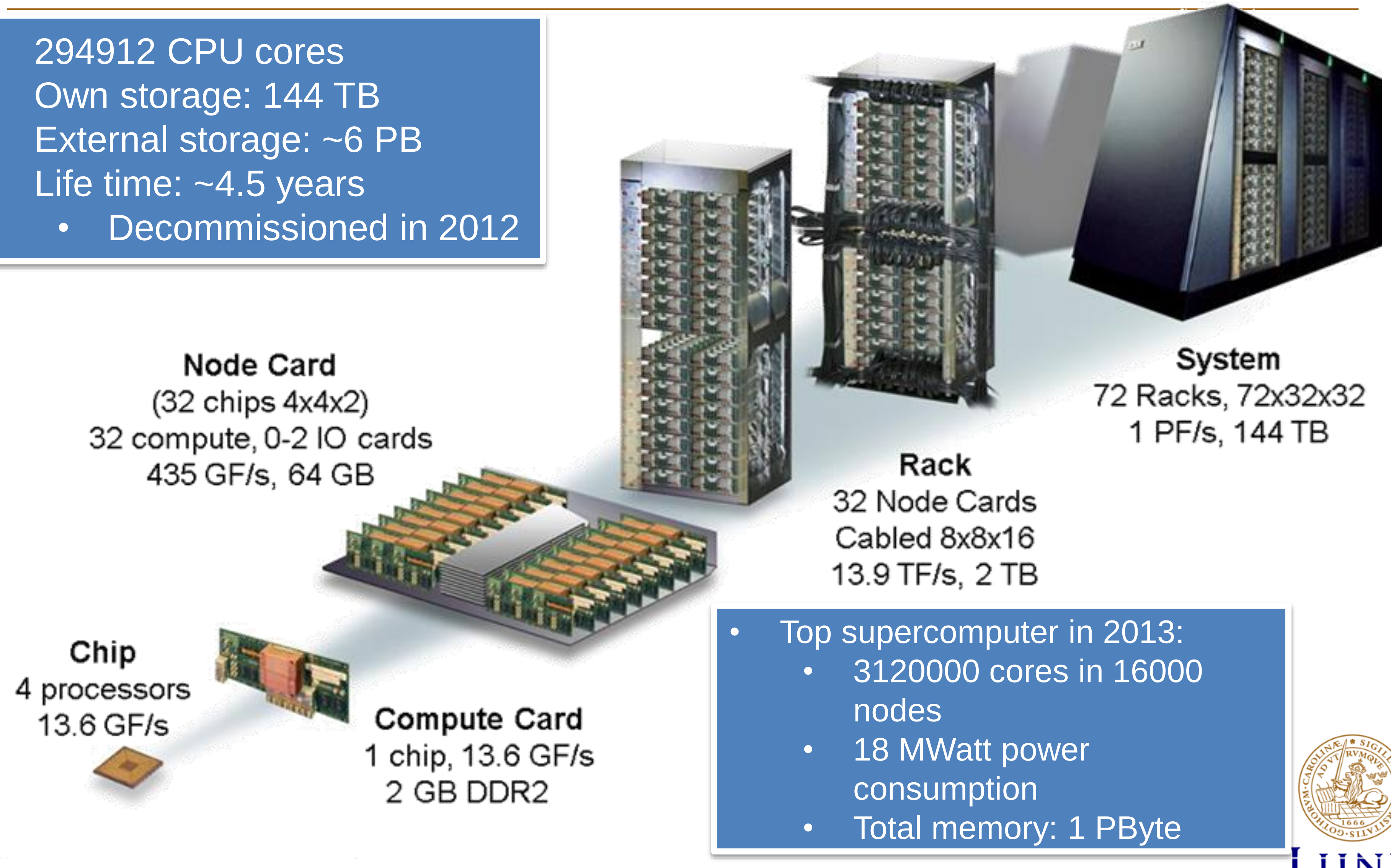
What is a computer? Brief summary

	Processor cores	Storage per core, TB	Operating system (typical)	Real	Virtual (“cloud”)
Personal computer (workstation)	$10^0 – 10^1$	$10^0 – 10^1$	Windows, MacOS, Linux	✓	✓
Cluster (farm)	$10^2 – 10^3$	$10^0 – 10^1$	Linux	✓	✓
Supercomputer	$10^4 – 10^6$	$10^{-3} – 10^{-1}$	Linux	✓	

- Deviations exist, boundaries are sometimes blurred
  - Fast interconnect between processors is a distinct property of supercomputers
- For the purposes of this course, such classification is enough

An old supercomputer: Blue Gene/P

- 294912 CPU cores
- Own storage: 144 TB
- External storage: ~6 PB
- Life time: ~4.5 years
    - Decommissioned in 2012



- Top supercomputer in 2013:
    - 3120000 cores in 16000 nodes
    - 18 MWatt power consumption
    - Total memory: 1 PByte

Graphics by IBM

LUND UNIVERSITY

Linux clusters

A very old traditional Linux cluster

The newest *Abisko* Linux cluster in Umeå

LUND UNIVERSITY

Computer memory

- In what follows, "memory" normally means primary memory – volatile, high-speed access
  - Mechanical non-volatile storage is slower, can be referred to as secondary memory, but we will call it storage (disks, flash memory etc)
  - Primary memory written to secondary memory is called virtual memory
- PCs and clusters have similar architectures memory-wise, while supercomputers share memory globally between cores
  - This is why supercomputers can not be virtualized
  - Memory in PCs and clusters can be shared programmatically

LUND UNIVERSITY

Memory modules

Memory in an HP server

Memory for a SUN blade server

High-performance memory for professionals

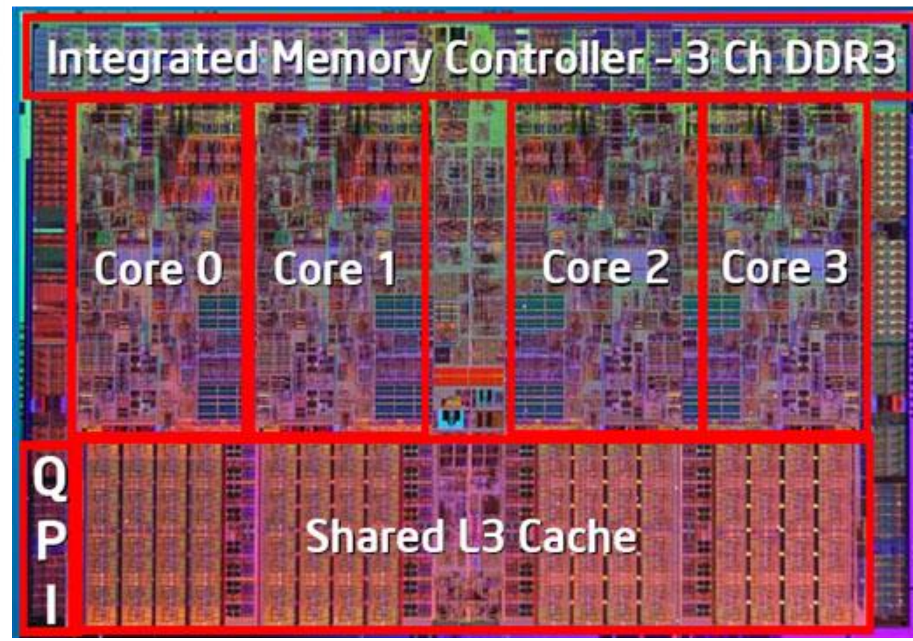
CPUs and cores

- CPU – Central Processing Unit – a chip that performs arithmetic, logical and input/output operations
- Modern chips usually contain several units and are referred to as <u>multicore</u> processors
  - Terminology is still confused: some call each processor a core, and the multicore chip – a CPU. Others call each core a CPU.
  - Cores on one chip usually share memory and input/output channel
- There are also GPUs – Graphical Processing Units – chips optimized to process graphics, good for parallel data processing

Processing Units

CPUs

Internals of a multicore CPU (Intel's Nehalem)

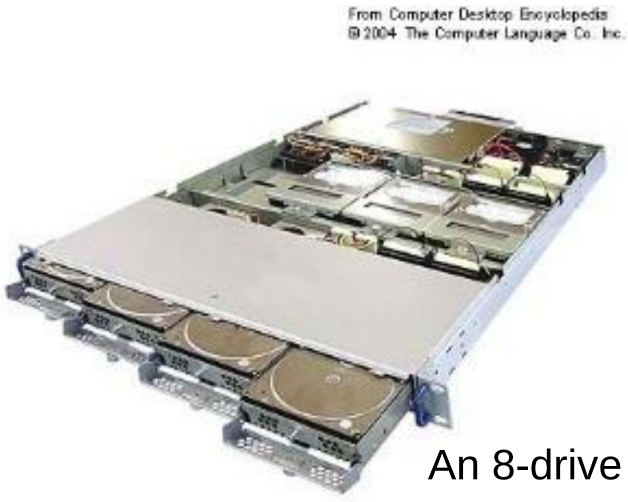


GPU

Storage

- Storage necessary for operating system, software and processing usually comes as disks close to CPUs
  - Diskless servers are also possible, though rare
- For permanent storage, dedicated disk servers are manufactured
  - Computing servers with very large storage capacity (dozens of Terabytes), optimized for fast access and back-up: low latency or on-line storage
- For archival, tape servers are used
  - Slow to access: serial read, require the tape to be fetched and inserted into the reading device: high latency or off-line storage

Storage servers



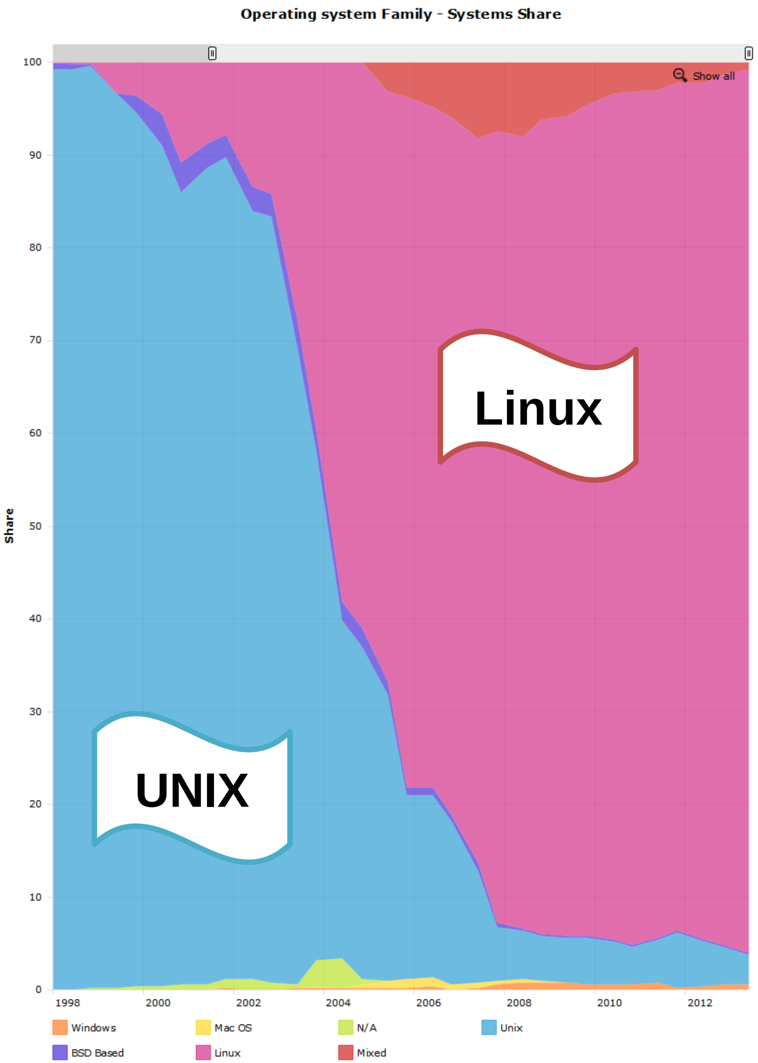
An 8-drive rack unit

Tape robot at FNAL

A disk storage rack fragment

Operating systems (OS)

- On PCs, Microsoft Windows dominates
  - For scientific computing, Linux and sometimes MacOS are used as well
- On clusters and supercomputers, Linux is by far dominant
  - Comes in many flavors
  - Often – RedHat Linux or its derivatives



LUND UNIVERSITY

Virtualization and Clouds

- Modern processors and operating systems allow full emulation of a computer
  - Such emulation is called virtualization
  - Everything is virtualized: CPU, network cards, disk partitions etc
  - Practical use: if your program works in one OS, and your PC uses another, you can simply emulate the computer with the necessary OS
    - » System to emulate is encapsulated in virtual images
    - » One real machine can host several virtual ones
- One can rent a virtual PC or even a virtual cluster from Cloud providers
  - Cloud servers are very large clusters, optimized to host virtual machines
  - Other Cloud services also exist: storage, databases etc

Scientific computing

APPROACHES AND TOOLS

Personal use – PCs, workstations

- Everybody likes to have one or two
- Powerful enough for many scientific tasks

- Strictly personal
- Heavily customized

LUND UNIVERSITY

Customized shared service – clusters, supercomputers

There is always demand and supply

- Systems are customized, but each can serve different users
- Disparate systems can be federated: create computing Grids

LUND UNIVERSITY

Generic service for rent – Clouds

Now exist for computing, data storage, databases etc

- Each Cloud is different, but each can be seemingly infinite because of virtualization
- Users can customize their rent
- No high performance

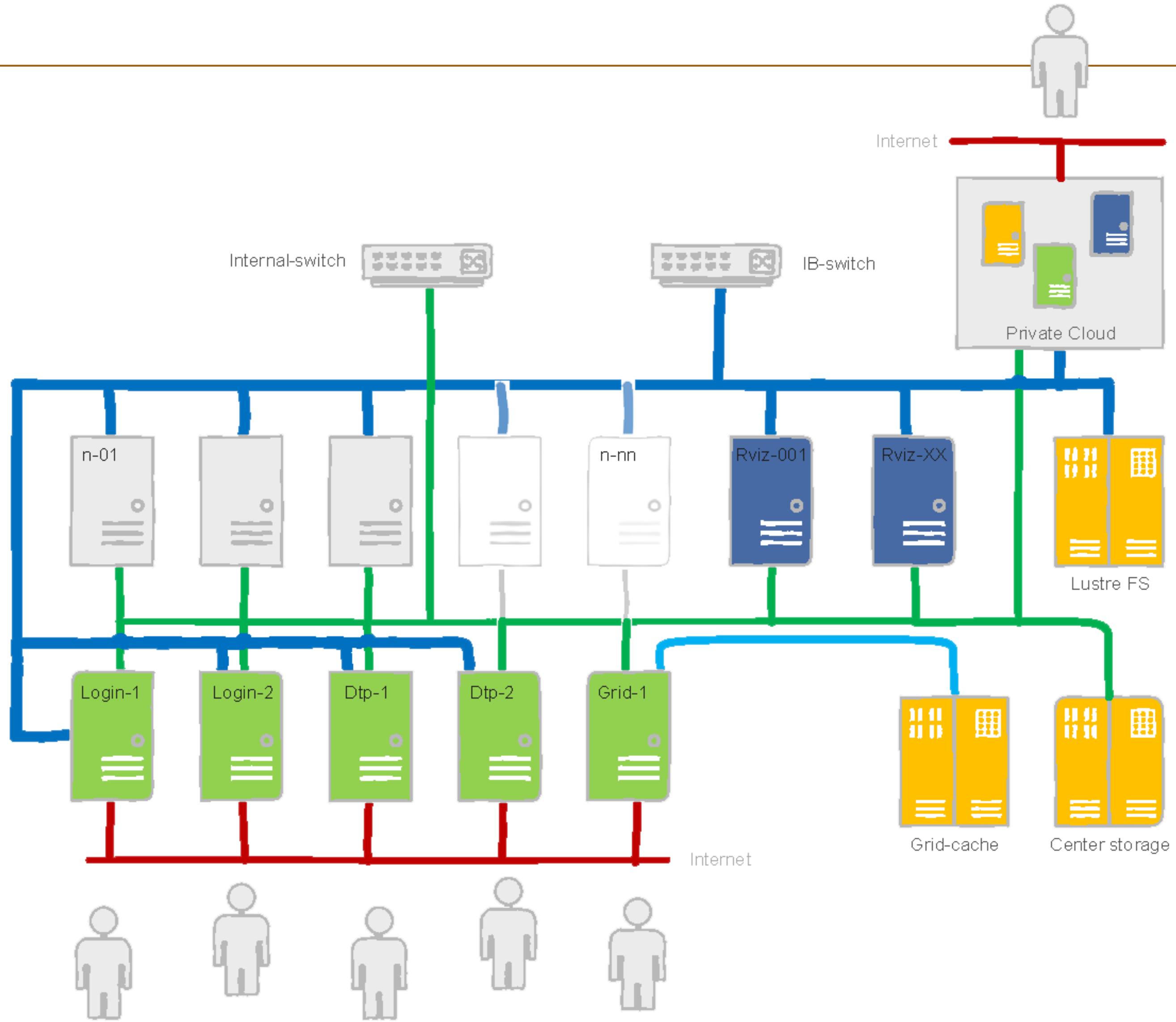
LUND UNIVERSITY

Our scope: clusters

- Computing facilities in universities and research centers usually are Linux clusters
  - Some supercomputer-grade facilities are actually clusters, too
- A cluster is a (comparatively) loosely coupled set of computing systems presented to users as a single resource
  - A typical cluster has a head node (or a few) and many worker nodes
    - » A node is a unit housing processors and memory
  - Distribution of load to worker nodes is orchestrated by means of Local Resource Management Systems (a.k.a. batch systems)
    - » Many batch systems exist on the market: PBS, SLURM, LSF, SGE/OGE etc
- Every cluster is a heavily customised resource built for a range of specific applications

LUND UNIVERSITY

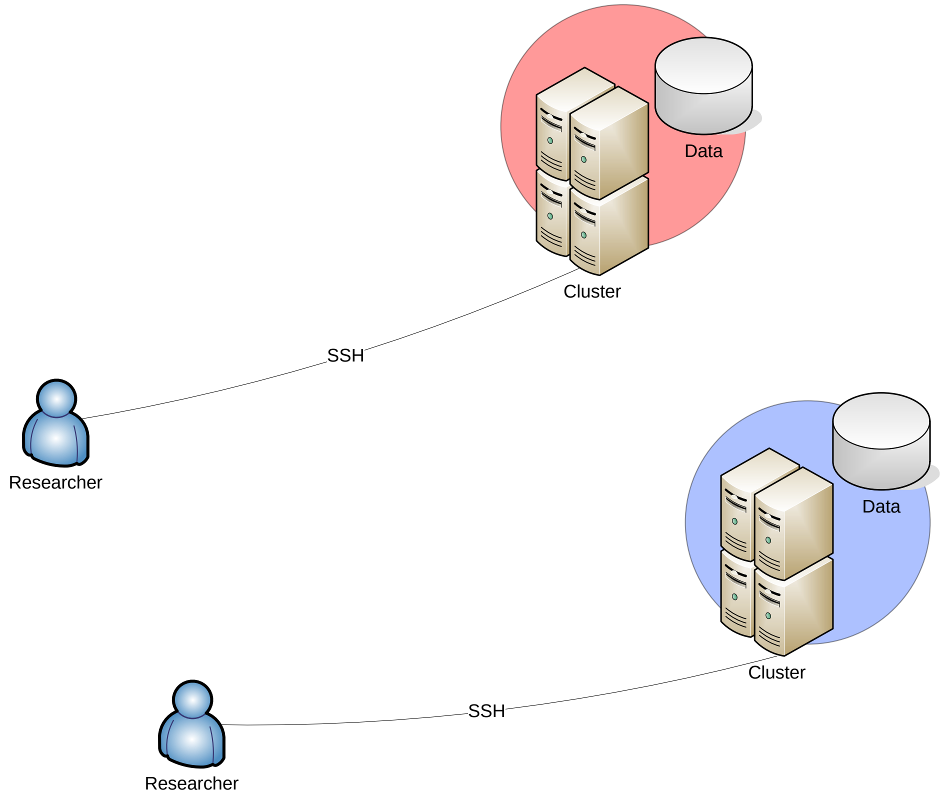
A possible future cluster at LUNARC



Graphics by J. Lindemann

LUND UNIVERSITY

Typical workflow on clusters



- Data placed in internal storage
- Users connect to the head node
- Specialised software is installed
  - Either centrally by admins, or privately by users
- Specialised scripts are used to launch tasks via batch systems
  - A single task is called a job
- A scientist usually has access to several clusters

Jobs and queues

- A batch system is software that schedules computational tasks to worker nodes according to given criteria and requirements
- A single unit of scheduling is called a **job**; some job requirements are:
  - A job can use a single core (serial job), or several cores at once (parallel job)
  - Consumes CPU time and astronomic (wall-clock) time
    - » A well-parallelized job will consume less wall time, but a comparable CPU time to a serial one
  - A job also consumes memory and disk space
  - A job may do intensive input/output operations (data processing)
  - A job may require public network connectivity
- When there are more jobs than resources, **queue** management is needed
  - A cluster may have several queues for different kinds of jobs (long, short, Grid etc)
  - A queue is actually a persistent partition of a cluster, exists even if there are no jobs

Scientific computing scenarios

A Infrequent tasks with moderate resource consumption
- E.g. Excel macros, simple image processing etc

B Large batches of similar (simple) independent tasks: serial jobs
- Processing of many images, analysis of accelerator collision events etc

C Lengthy resource-consuming tasks: often parallel jobs
- Pattern recognition, complex system simulation, parameter scanning, lattice QCD etc
  - Parallel jobs share memory, or exchange information by other means

Distributed computing motivations

- How to deal with increasing computing power and storage requirements?
  - For parallel jobs: buy larger clusters/supercomputers - $$$
  - For serial jobs: distribute them across all the community resources
    - » Preferably using the same access credentials
    - » The results must be collected in one place
    - » Progress needs to be monitored
    - » Uniform environment is also needed

Some Grid precursors

Distributed file systems: AFS, NFS4

- First implementation in ~1984
- Allow different systems to have common storage and software environment

Condor/HTCondor pools

- High Throughput Computing across different computers
- Started in ~1988 by pooling Windows PCs
- A variant often used as a cluster batch system

Networked batch systems: LSF, SGE

- Can use single batch system on many clusters since ~1994
- Variants of regular cluster batch systems

Volunteer computing: SETI@HOME, BOINC

- Target PC owners since ~1999
- Supports only a pre-defined set of applications

Grid concept – formulated in ~1999

Abstracted interfaces from systems

- No need for common batch systems or common file systems

Introduced security infrastructure

- Single sign-on
- Certificate-based authentication and authorisation

Introduced resource information system

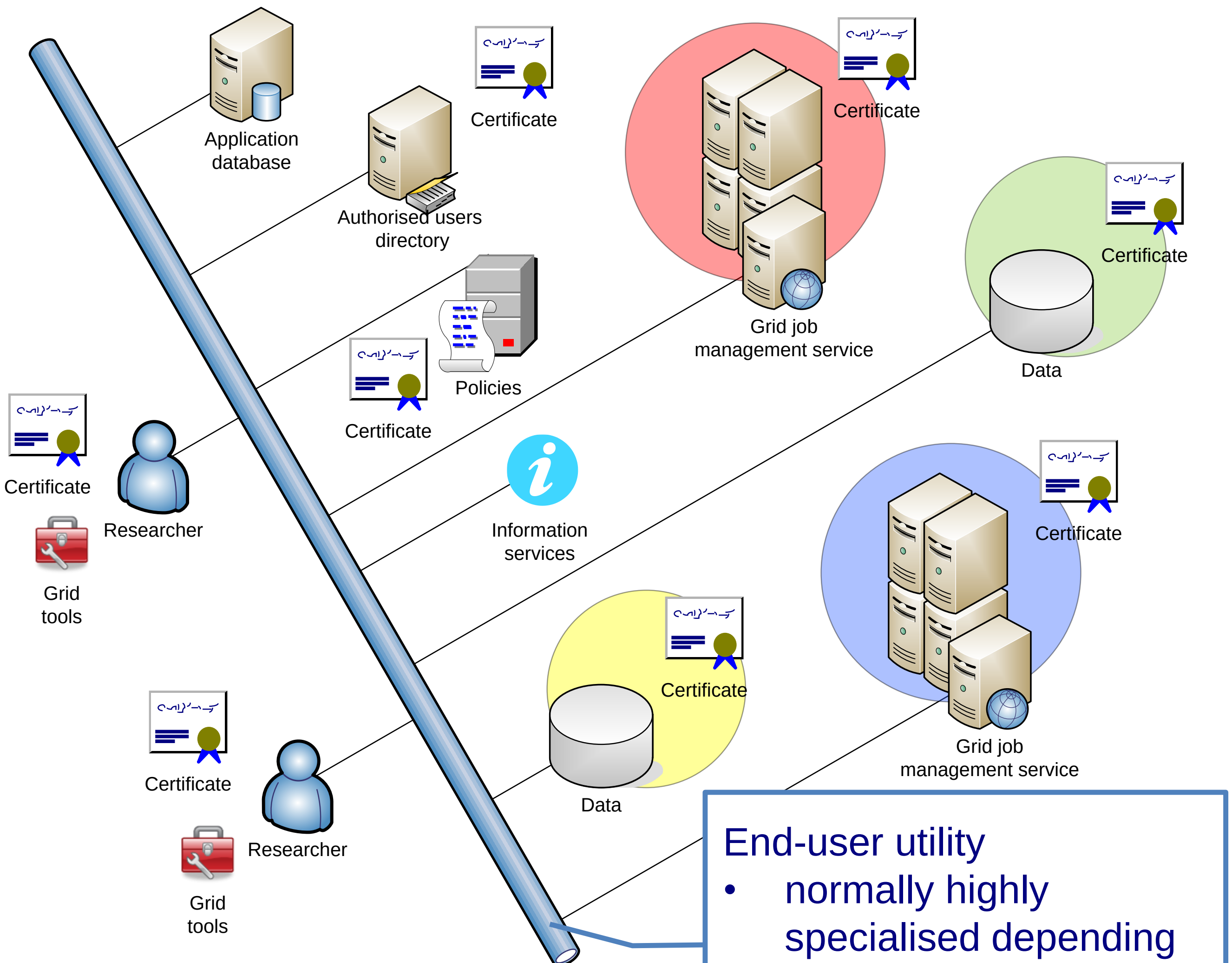
- Necessary for batch-like job management

Ideal for distributed serial jobs

- Initially was thought to be suitable even for parallel jobs

Grid is a technology enabling federations of heterogeneous conventional systems, facilitated by fast networks and a software layer that provides single sign-on and delegation of access rights through common interfaces for basic services

Overview of generic Grid components



End-user utility

- normally highly specialised depending on research task

Some Grid software providers

The first: Globus Toolkit (stems from the USA)
http://toolkit.globus.org/toolkit

- Provides computing capacity, basic storage capacity, and corresponding client tools
- Comes with extensive libraries and API, used by other providers
  - Especially for the Grid Security Infrastructure

Used in this course: ARC by NorduGrid
http://www.nordugrid.org/arc

- Provides computing capacity and client tools for job and file operations
- Developed in Lund, among other places

European Grid stack: EMI
http://www.eu-emi.eu

- Includes ARC along with many other components and services for storage, accounting, information, security etc

Exercise: cluster batch system

- Goal: understand basic concepts and explore inner works of a cluster through its batch system
  - Cluster: *Iridium* cluster at LUNARC
  - Batch system: SLURM
    - » https://computing.llnl.gov/linux/slurm/quickstart.html
- Steps:
  - Log in to the cluster:
    - » **`> ssh -X <username>@<clustername>`**
  - Inspect CPU and memory details:
    - » **`cat /proc/cpuinfo; cat /proc/meminfo; top`**
  - Check man pages for SLURM commands:
    - » **`sbatch, sinfo, srun, squeue, scontrol`**

Exercise (continued)

- List SLURM queues (partitions)
  - **`> sinfo`**
- Create file **`myscript`** (use provided examples)
- Submit jobs and check their status:
  - **`> sbatch -N4 myscript`**
  - **`> cat slurm-<jobid>.out`**
  - **`> squeue`**
  - **`> scontrol show job <jobid>`**
- Repeat with other examples
  - In a multi-core advanced example, pay attention how jobs are distributed across nodes and cores

Possible file `myscript`:

```
#!/bin/sh
#SBATCH -J "My job"
#SBATCH --time=1
srun hostname |sort
sleep 5m
```


LUND UNIVERSITY

Homework

- Describe your computing task using the terms defined today
  - Chose any of the tasks you currently do
  - If none exist, think of what kind of task you may need
- Maximum one page
- Submit by e-mail by April 16

- Bring along a USB memory key for the next lecture!
  - Needed to store your private certificate

LUND UNIVERSITY