Large Scale in What Dimension?

Miron Livny
Center for High Throughput Computing
Computer Sciences Department
University of Wisconsin-Madison

It does (not) scale!

What does it mean and should you care?

Scale

Verb

1. measure by or as if by a scale; "This bike scales only 25 pounds"
2. pattern, make, regulate, set, measure, or estimate according to some rate or standard
3. take by attacking with scaling ladders; "The troops scaled the walls of the fort"
4. reach the highest point of; "We scaled the Mont Blanc"
5. climb up by means of a ladder
6. remove the scales from; "scale fish"
7. measure with or as if with scales; "scale the gold"
8. size or measure according to a scale; "This model must be scaled down"

WordNet® 3.0, © 2006 by Princeton University.

From Wikipedia, the free encyclopedia

In telecommunications and software engineering, **scalability** is a desirable property of a system, a network, or a process, which indicates its ability to either handle growing amounts of work in a graceful manner, or to be readily enlarged. For example, it can refer to the capability of a system to increase total throughput under an increased load when resources (typically hardware) are added. An analogous meaning is implied when the word is used in a commercial context, where scalability of a company implies that the underlying business model offers the potential for economic growth within the company.

Scalability, as a property of systems, is generally difficult to define and in any particular case it is necessary to define the specific requirements for scalability on those dimensions which are deemed important. It is a highly significant issue in electronics systems, database, routers, and networking. A system whose performance improves after adding hardware, proportionally to the capacity added, is said to be a **scalable system**. An algorithm, design, networking protocol, program, or other system is said to **scale** if it is suitably efficient and practical when applied to large situations (e.g. a large input data set or large number of participating nodes in the case of a distributed system). If the design fails when the quantity increases then it **does not scale**.

Food for thought

- What dimensions do you care about?
- How far do you want/need to scale up (down?) the system in each of these dimensions?
- What are the tradeoffs between scalability and other system properties like time to market or cost of ownership

Dimensions

- Number of physical components – CPUs, disks, networks, …
- Number of logical entities – messages, transactions, jobs, …
- Number of users
- Number of customers
- Number of applications
- Number of sites
- Number of reliability 9's
- Number of organizations
- Number of software developers
- Number of platforms
- Number of releases
- Number of languages
- …

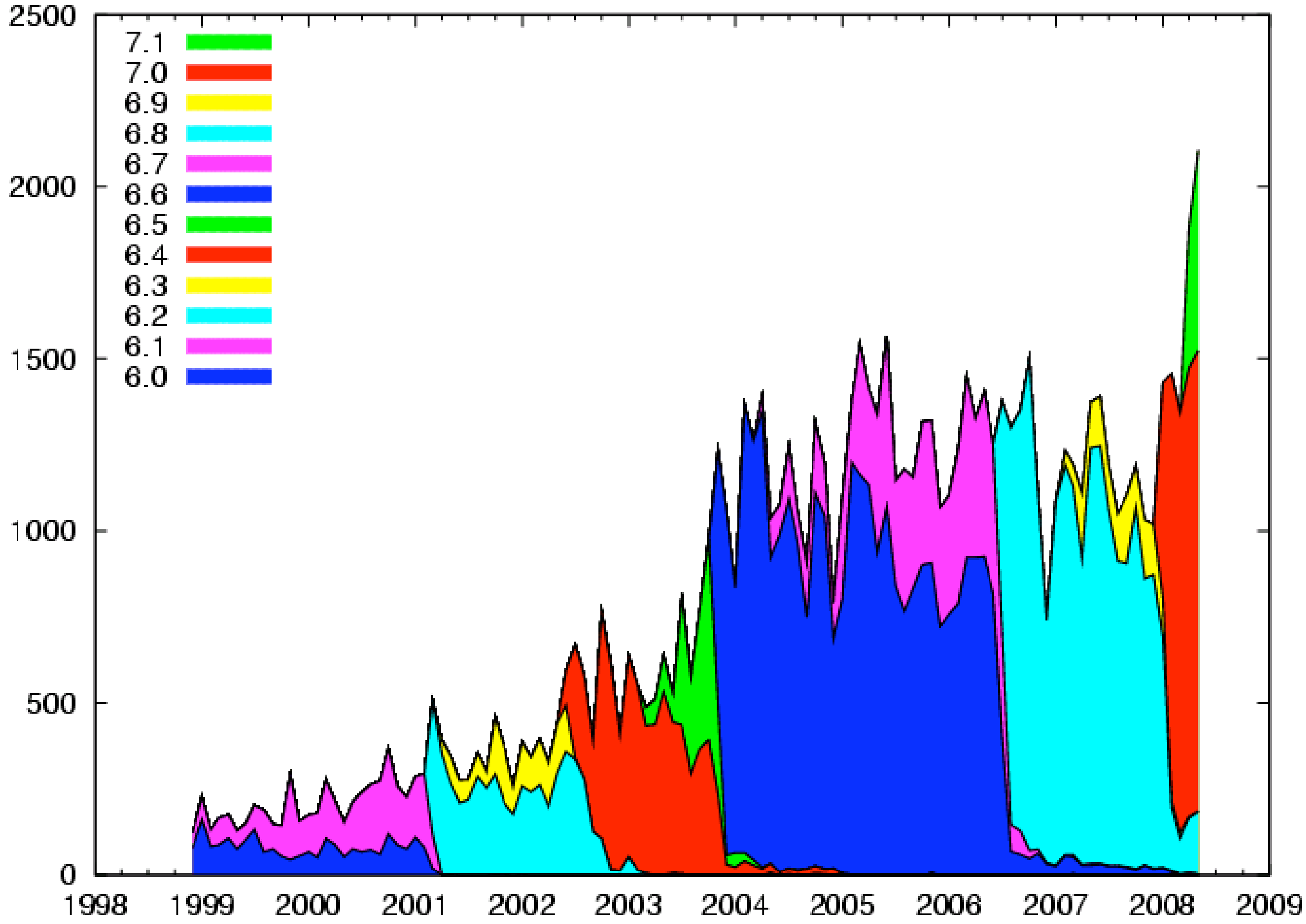
Condor Team 2007

The Condor Project (Established '85)

Distributed Computing **research** performed by a team of ~**35** faculty, full time staff and students who

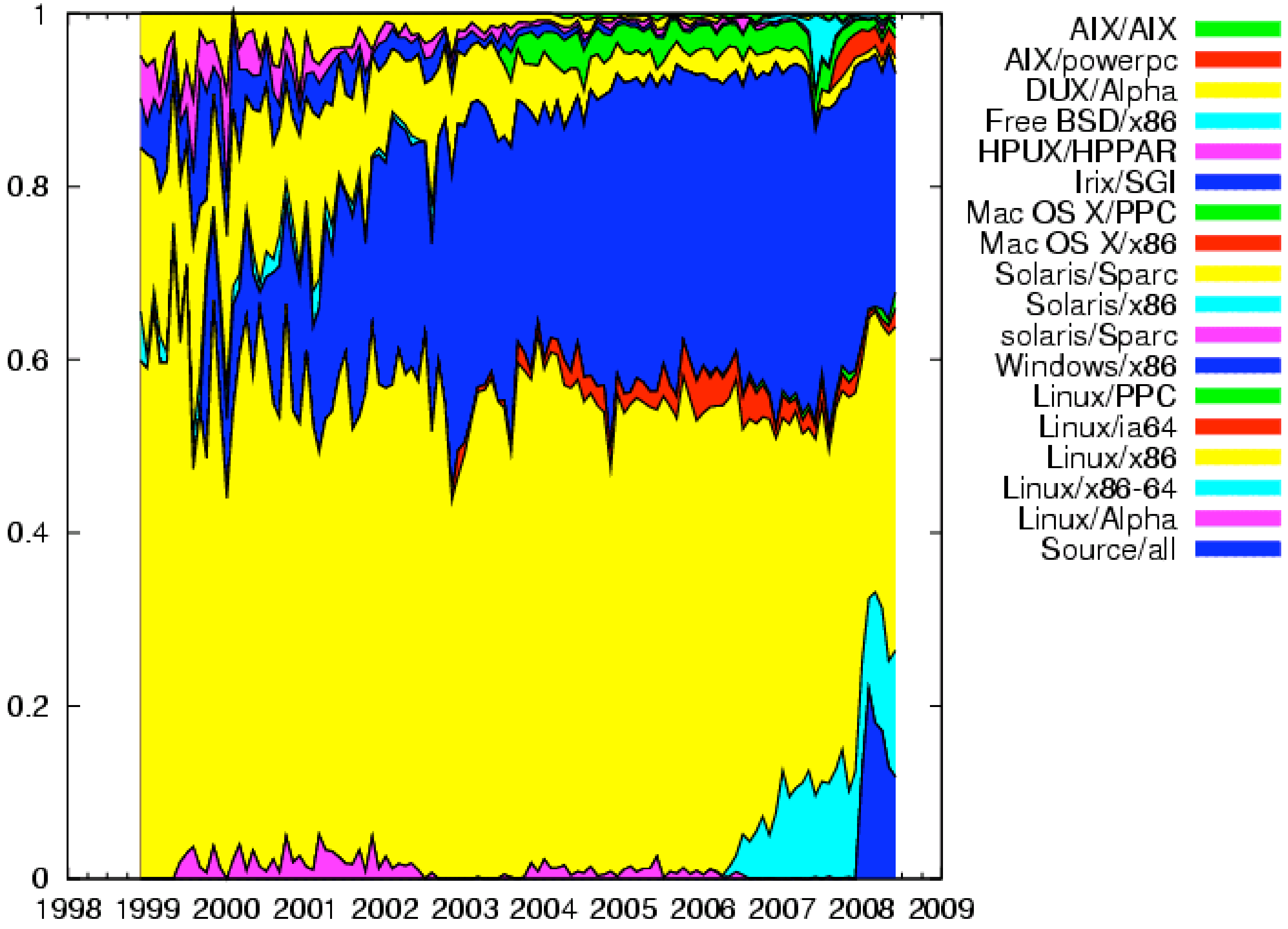
- face **software/middleware engineering** challenges in a UNIX/Linux/Windows/OS X environment,
- involved in national and international **collaborations**,
- interact with **users** in academia and industry,
- maintain and support a distributed **production** environment (more than 5000 CPUs at UW),
- and educate and train **students**.

Condor Monthly Downloads



OS and Architecture Fractional

(built on 17 and tested on 32)



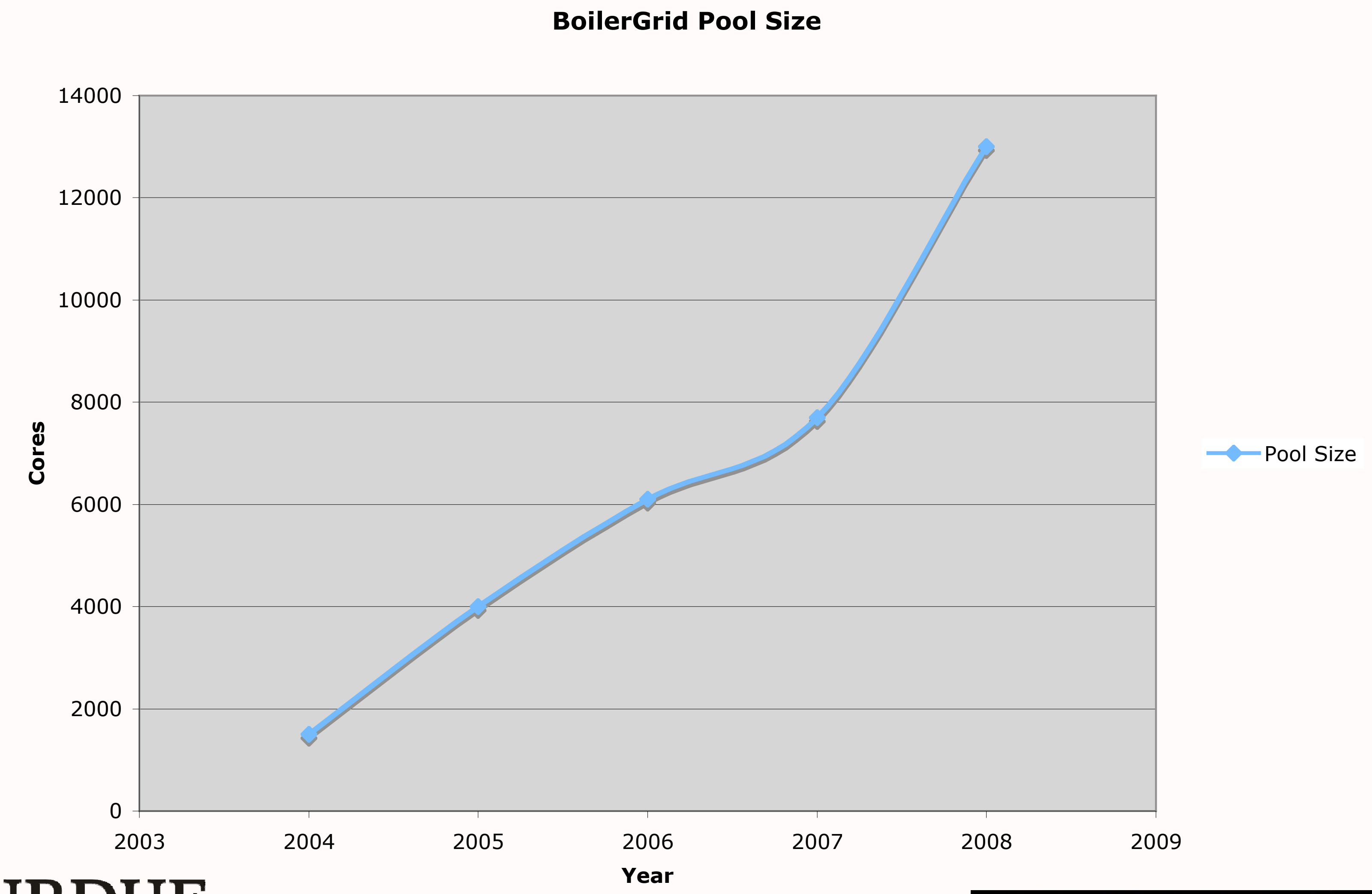
BoilerGrid

- Purdue Condor Grid (BoilerGrid)
  - Comprised of Linux HPC clusters, student labs, machines from academic department, and Purdue regional campuses
- **8900** batch slots today..
- **14,000** batch slots in a few weeks
- 2007 - Delivered over 10 million CPU-hours to high-throughput science to Purdue and national community through Open Science Grid and TeraGrid

BOILER GRID

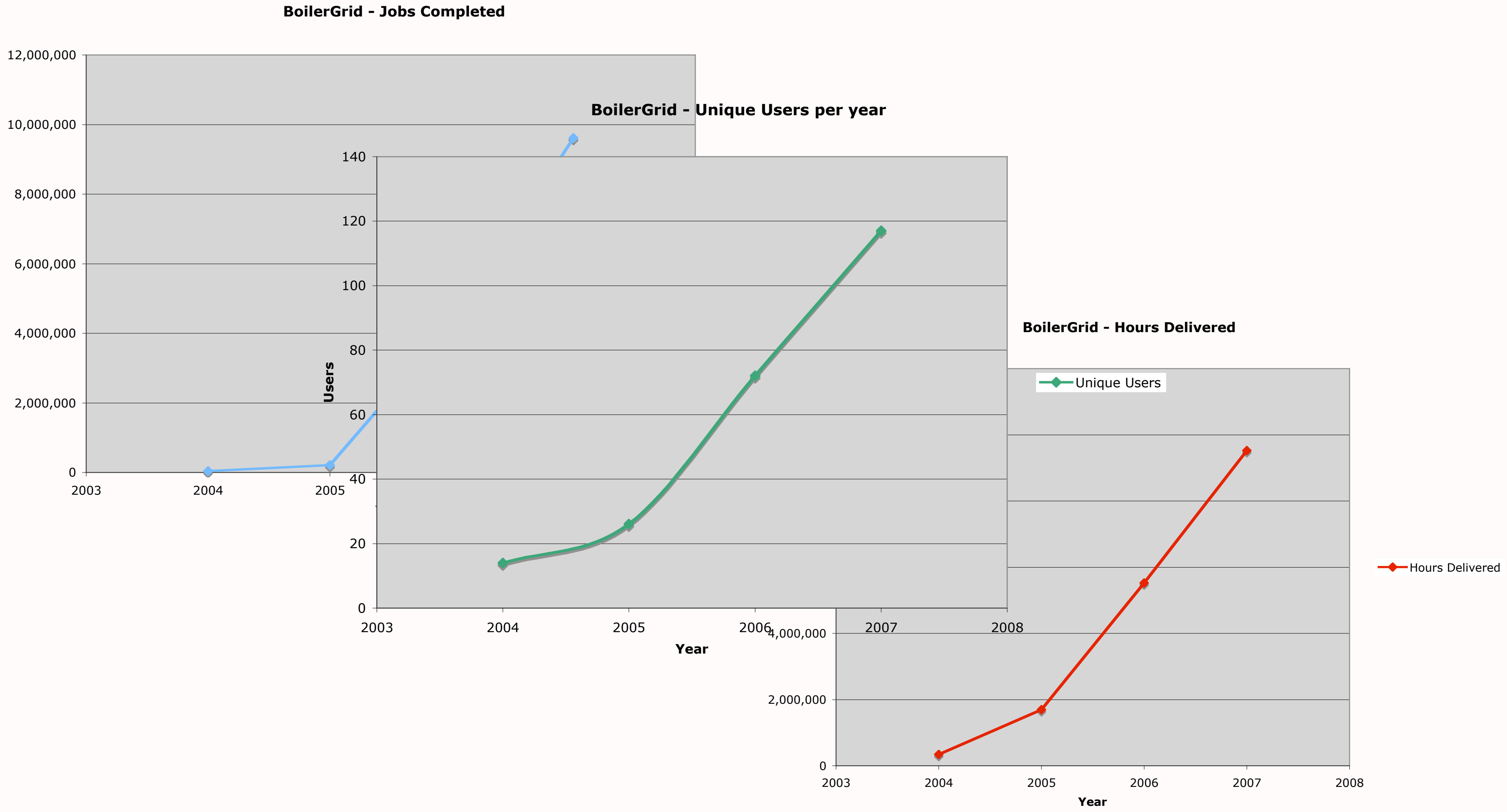
Rosen Center for Advanced Computing

BoilerGrid - Growth



Rosen Center for Advanced Computing

BoilerGrid - Results



BoilerGrid - Jobs Completed

BoilerGrid - Unique Users per year

BoilerGrid - Hours Delivered

PURDUE UNIVERSITY

BOILER GRID

The search for SUSY

Sanjay Padhi is a UW Chancellor Fellow who is working at the group of Prof. Sau Lan Wu at CERN

Using Condor Technologies he established a “grid access point” in his office at CERN

Through this access-point he managed to harness in 3 month (12/05-2/06) **more than 500 CPU years** from the LHC Computing Grid (LCG) the Open Science Grid (OSG) and UW Condor resources

The Laser Interferometer Gravitational-Wave Observatory

Supported by the United States National Science Foundation

Use of Condor by the LIGO Scientific Collaboration

Gregory Mendell, LIGO Hanford Observatory

On behalf of the LIGO Scientific Collaboration

Use of Condor by the LIGO Scientific Collaboration

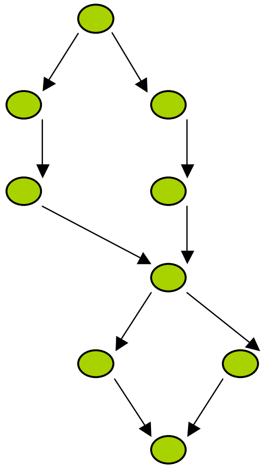
Condor High Throughput Computing

- Condor handles 10’s of millions of jobs per year running on the LDG, and up to 500k jobs per DAG.

- Condor standard universe checking pointing widely used, saving us from having to manage this.

- At Caltech, 30 million jobs processed using 22.8 million CPU hrs. on 1324 CPUs in last 30 months.

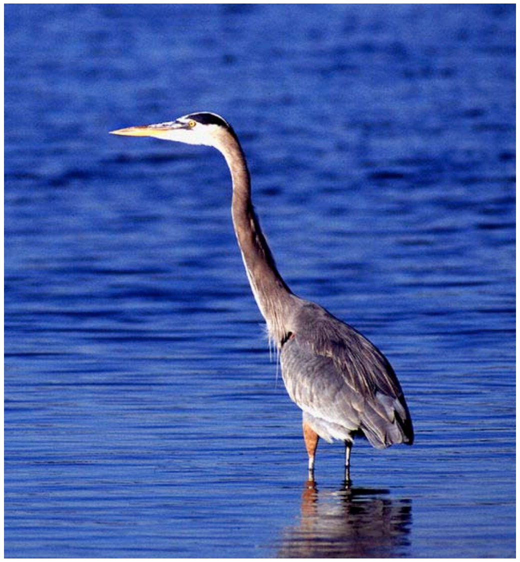
- For example, to search 1 yr. of data for GWs from the inspiral of binary neutron star and black hole systems takes ~2 million jobs, and months to run on several thousand ~2.6 GHz nodes.



What is the Blue Heron Project?

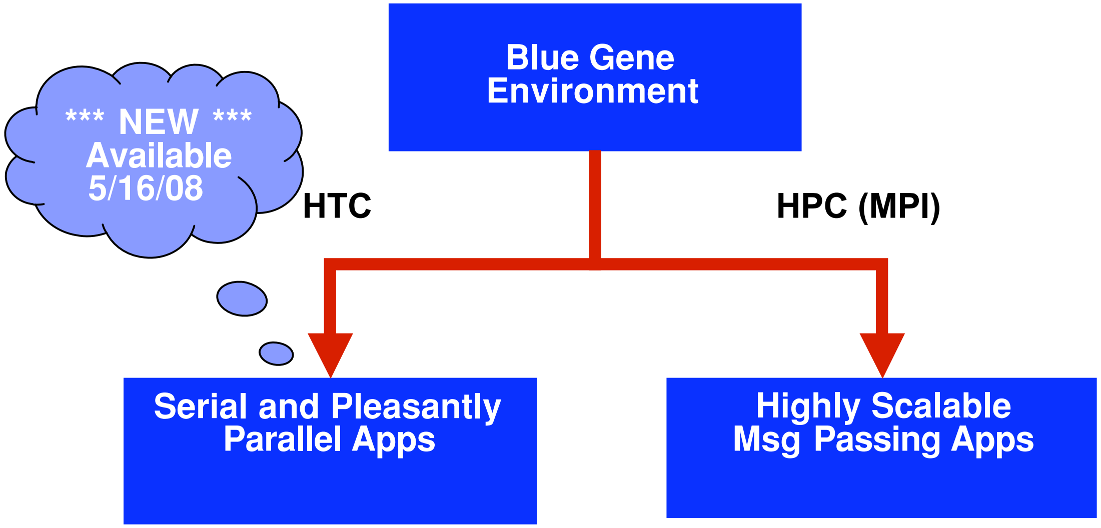
Blue Heron = Blue Gene/P HTC and Condor

Blue Heron provides a complete integrated solution that gives users a simple, flexible mechanism for submitting single-node jobs.



- ➢ Blue Gene looks like a "cluster" from an app’s point of view
- ➢ Blue Gene supports hybrid application environment
  - ▪ Classic HPC (MPI/OpenMP) apps and now HTC apps

Paths Toward a General Purpose Machine


*** NEW ***
Available
5/16/08
Blue Gene Environment
HTC
HPC (MPI)
Serial and Pleasantly Parallel Apps
Highly Scalable Msg Passing Apps

What will Red Hat be doing?

Red Hat will be investing into the Condor project locally in Madison WI, in addition to driving work required in upstream and related projects. This work will include:

- Engineering on Condor features & infrastructure
  - Should result in tighter integration with related technologies
  - Tighter kernel integration
- Information transfer between the Condor team and Red Hat engineers working on things like Messaging, Virtualization, etc.
- Creating and packaging Condor components for Linux distributions
- Support for Condor packaged in RH distributions

All work goes back to upstream communities, so this partnership will benefit all.

- Shameless plug: *If you want to be involved, Red Hat is hiring...*

How are we working together.

We work along side of the Condor team

- Always do what is best for the Condor project – community first!
- Have both on-campus resources and remote, and yes we are hiring

What areas have we working on so far with UW

- Build systems
- Libvirt (native Linux virtualization support – via gap)
- EC2 – more presentations on that
- Using hooks / Low Latency scheduling
- Management

Condor team also get access to make changes in Linux upstream through collaboration

- Expect more happening in this area in future.

Fedora Nightlife

› **About Nightlife**

Fedora Nightlife is a project for creating a Fedora community grid. People will be able to donate idle capacity from their own computers to an open, general-purpose Fedora-run grid for processing socially beneficial work and scientific research that requires access to large amounts of computing power. Given the large number of Fedora users, we hope to eventually be able to build a community grid of over a million nodes at Fedora. This will be a great example of the power of the Fedora community, give people new and meaningful ways to contribute to Fedora, advance the development of large-scale grid software, and lead to real benefits for the world.

A core part of Fedora Nightlife's mission will be to provide a complete open-source grid infrastructure suitable for computing a full spectrum of tasks. This includes:

The core operating system and execution environments
The necessary libraries for many workloads
The workload scheduler
Facilities for managing and enforcing various policies
and so on

Fedora Nightlife will leverage the Condor project, which was created and hosted by the University of Wisconsin Madison, for scheduling and harnessing donated computing power. Last year, Red Hat and the University of Wisconsin signed a strategic partnership around Condor. Part of this partnership entailed releasing Condor's source code under an OSI-approved open source license. **As a result, we now have Condor packaged at Fedora, and upstream development continues to happen at the University of Wisconsin repository in an open manner.**

How to select the scalability "Sweet Spot"?

Rule #1

A system that works and does not scale is always preferable over a system that scales but does not work

- Building a system that works forces you to cover all the bases. Once they are covered you can address scalability
- You can not address scalability without users and can not have users without a working system

Rule #2

Address functionality and deemphasize performance

- η A 10x improvement changes the functionality of your system
- η Performance enhancements are likely to introduce scalability obstacles

Rule #3

Use as much as you can proven widely adopted technologies

η Everything that can go wrong will go wrong in a large system

η Debugging does not scale

η Minimizes requirements on hosting environment

Rule #4

Understand weaknesses, limitations and vulnerabilities

- η No free lunches
- η Under promise and over deliver
- η Fewer blind spots

Rule #5

Eat your own dog food and treasure your customers

- η Keeps you honest
- η Fix/debug system in a friendly environment
- η Guide development
- η Understand tradeoffs

Rule #6

Try to understand what makes physical, biological and social systems/organizations scalable

η Identify the scalability principals

η Identify the tradeoffs

η Identify the limitations

η Study the history/evolution

Designing for Scalability

Claims for “benefits” provided by Distributed Processing Systems

- High Availability and Reliability
- High System Performance
- Ease of Modular and Incremental Growth
- Automatic Load and Resource Sharing
- Good Response to Temporary Overloads
- Easy Expansion in Capacity and/or Function

“What is a Distributed Data Processing System?” , P.H. Enslow, Computer, January 1978

Definitional Criteria for a Distributed Processing System

η Multiplicity of resources

η Component interconnection

η **Unity of control**

η System transparency

η **Component autonomy**

P.H. Enslow and T. G. Saponas ***“”Distributed and Decentralized Control in Fully Distributed Processing Systems”*** **Technical Report, 1981**

Unity of Control

All the component of the system should be **unified** in their desire to achieve a **common goal**. This goal will determine the rules according to which each of these elements will be controlled.

Component autonomy

The components of the system, both the logical and physical, should be **autonomous** and are thus afforded the ability to refuse a request of service made by another element. However, in order to achieve the system’s goals they have to interact in a **cooperative** manner and thus adhere to a common set of policies. These policies should be carried out by the control schemes of each element.

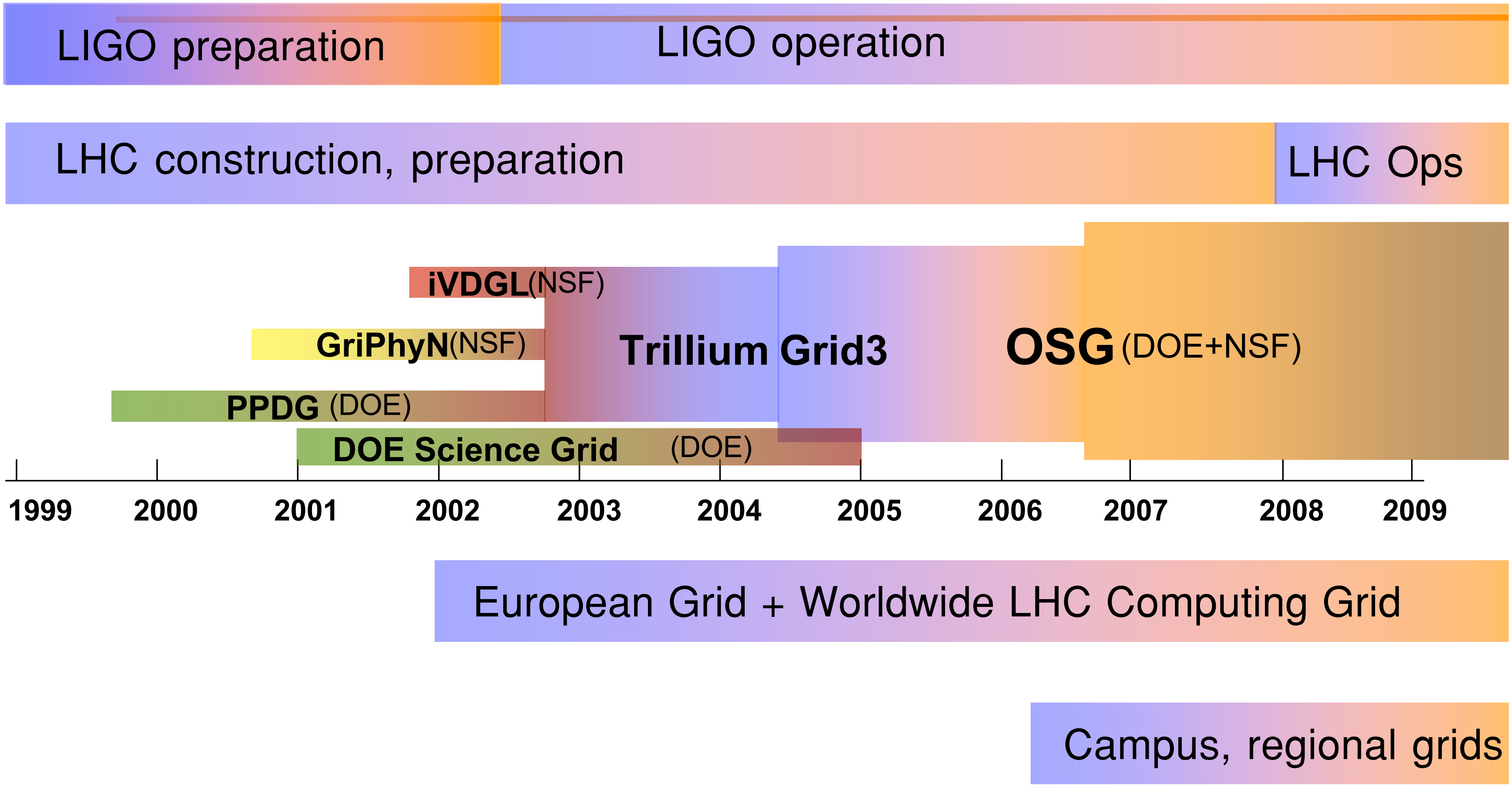
Open Science Grid (OSG) vision

Transform processing and **data** intensive science through a cross-domain self-managed **national distributed** cyber-infrastructure that **brings together** campus and community infrastructure and facilitating the needs of **Virtual Organizations** (VO) at all scales

]

Open Science Grid

The Evolution of the OSG



LIGO preparation

LIGO operation

LHC construction, preparation

LHC Ops

iVDGL(NSF)

GriPhyN(NSF)

PPDG (DOE)

DOE Science Grid (DOE)

Trillium **Grid3**

OSG (DOE+NSF)

European Grid + Worldwide LHC Computing Grid

Campus, regional grids

]

Open Science Grid

The OSG Consortium

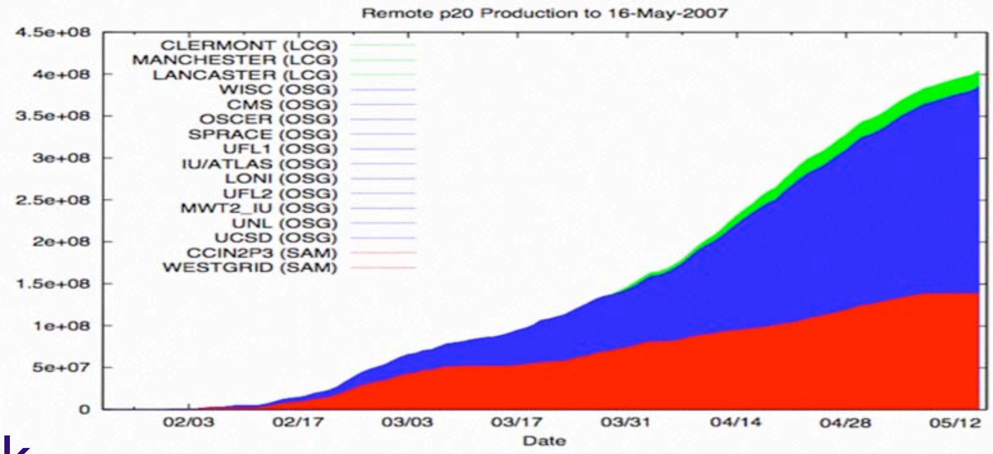
- **> 20 Scientific Virtual Organizations:** LHC, STAR, LIGO, NanoHub etc.
- **> 25 Resource Providers:** DOE National Labs, University Facilities etc.
  - 10 Storage focused resources
- **>10 Software Providers (including External Projects):** Condor, Globus, Storage Resource Manager, Internet2, ESNET, CEDPS, Fermilab Accounting etc.
- **>4 Partners - Ex-Officio:** EGEE, TeraGrid, NWICG, TIGRE, APAC etc.

D0 Data Re-Processing

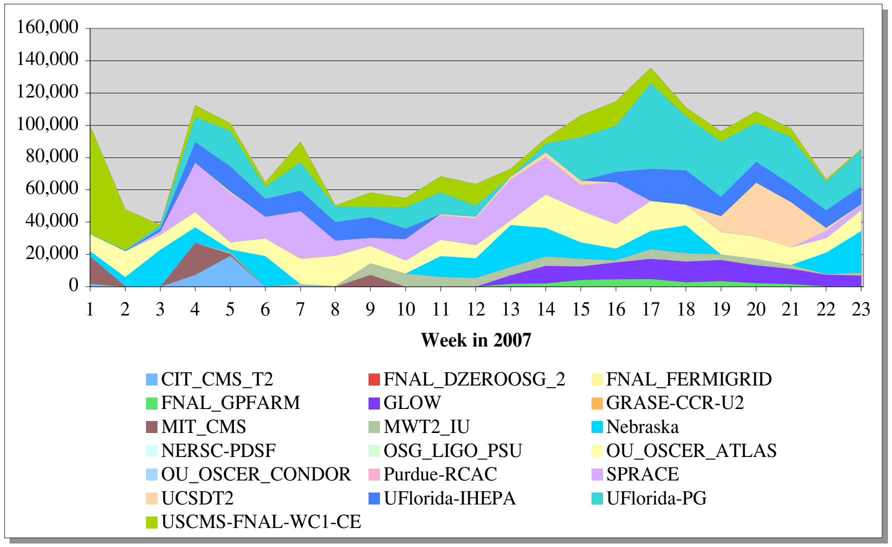
Open Science Grid

12 sites contributed up to 1000 jobs/day

Total Events



OSG CPUHours/Week

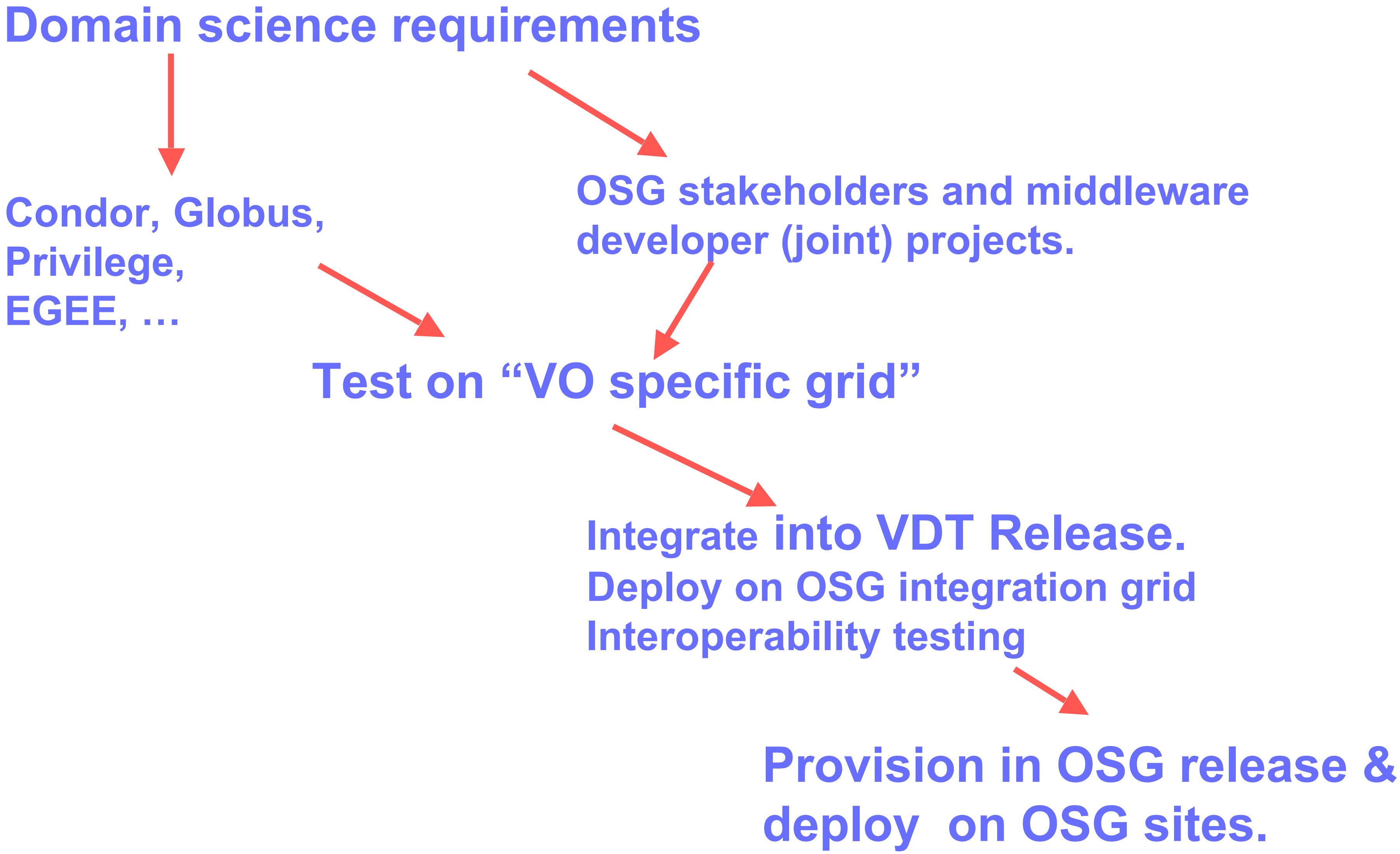


2M	CPU hours
286M	events
286K	Jobs on OSG
48TB	Input data
22TB	Output data

OSG challenges

- Develop the **organizational** and management structure of a consortium that drives such a Cyber Infrastructure
- Develop the **organizational** and management structure for the project that builds, operates and evolves such Cyber Infrastructure
- Maintain and evolve a **software stack** capable of offering powerful and dependable capabilities that meet the science objectives of the NSF and DOE scientific communities
- Operate and evolve a dependable and well managed **distributed facility**

OSG Middleware Evolution



Domain science requirements

Condor, Globus,
Privilege,
EGEE, …

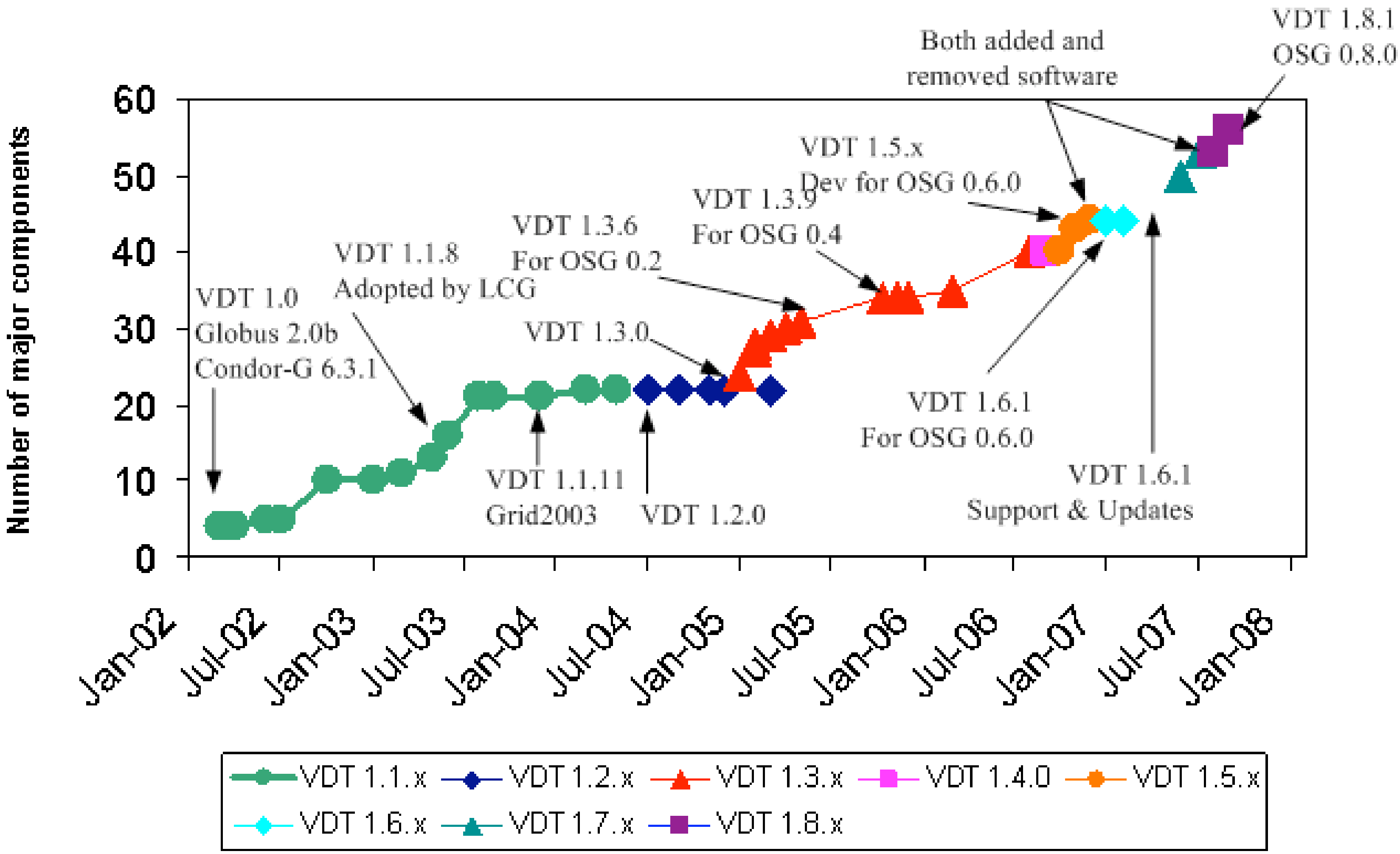
OSG stakeholders and middleware
developer (joint) projects.

Test on “VO specific grid”

Integrate into VDT Release.
Deploy on OSG integration grid
Interoperability testing

Provision in OSG release &
deploy on OSG sites.

How much software?



Design Principal #1

Maximize local autonomy and regulate consumption of all local resources

η Use “circuit breakers” to prevent overheating of resources

η Use locally defined policies

η Be skeptical about all information offered by external sources

Design Principal #2

Work delegation and resource allocation protocols should be viewed as a “lease” and provide effective exception handling capabilities

- η One of the parties may “disappear”
- η One of the parties may “fail” to deliver the goods
- η Quality of feedback determines recovery options

Design Principal #3

Allow clients to “turn around” and act as servers and servers as clients

- η Enables hierarchical structures through “recursion”
- η Reduces number of elements/abstractions/operstors in the system

Design principal #4

Treat providers and consumers of resources symmetrically (as much as you can)

- η Eliminates historical/temporary biases
- η Exposes weakness in the design
- η Challenges basic/default assumptions

Design Principal #5

Never underestimate the applicability validity of “motherhood and apple-pie”

η Less is more

η Simple is beautiful

η Sharing is caring

η Easier said then done

η Order of magnitude **do** matter

High Throughput Computing

I first introduced the distinction between High Performance Computing (HPC) and High Throughput Computing (HTC) in a seminar at the NASA Goddard Flight Center in July of **1996** and a month later at the European Laboratory for Particle Physics (CERN). In June of 1997 HPCWire published an interview on High Throughput Computing.

Why HTC?

For many experimental scientists, scientific progress and quality of research are strongly linked to computing throughput. In other words, they are less concerned about instantaneous computing power. Instead, what matters to them is the amount of computing they can harness over a month or a year --- they measure computing power in units of scenarios per day, wind patterns per week, instructions sets per month, or crystal configurations per year.

High Throughput Computing is a 24-7-365 activity

$$\text{FLOPY} \neq (60*60*24*7*52)*\text{FLOPS}$$

Every Community can benefit from the services of

Matchmakers!

eBay is a matchmaker

Why? Because ...

.. someone has to bring together community members who have requests for goods and services with members who offer them.

- Both sides are looking for each other
- Both sides have constraints/requirements
- Both sides have preferences

Being a Matchmaker

- Symmetric treatment of all parties
- Schema "neutral"
- Matching policies defined by parties
- "Just in time" decisions
- Acts as an "advisor" not "enforcer"
- Can be used for "resource allocation" and "job delegation"

“ **…** We claim that these **mechanisms**, although originally developed in the context of a cluster of workstations, are also applicable to computational **grids**. In addition to the required flexibility of services in these grids, a very important concern is that the system be **robust** enough to run in “**production mode**” continuously even in the face of component failures. **…** “

Miron Livny & Rajesh Raman, ***"****High Throughput Resource Management****"*****, in “ *The Grid: Blueprint for a New Computing Infrastructure*”.**

In the words of the CIO of Hartford Life

"Resource: What do you expect to gain from grid computing? What are your main goals?

Severino: Well number one was scalability. …

Second, we obviously wanted **scalability with stability**. As we brought more servers and desktops onto the grid we didn't make it any less stable by having a bigger environment.

The third goal was cost savings. One of the most …"

Grid Computing At Hartford Life -*From Resource, February 2005 Copyright by LOMA*

How can we accommodate an **unbounded** need for computing with an **unbounded** amount of resources?