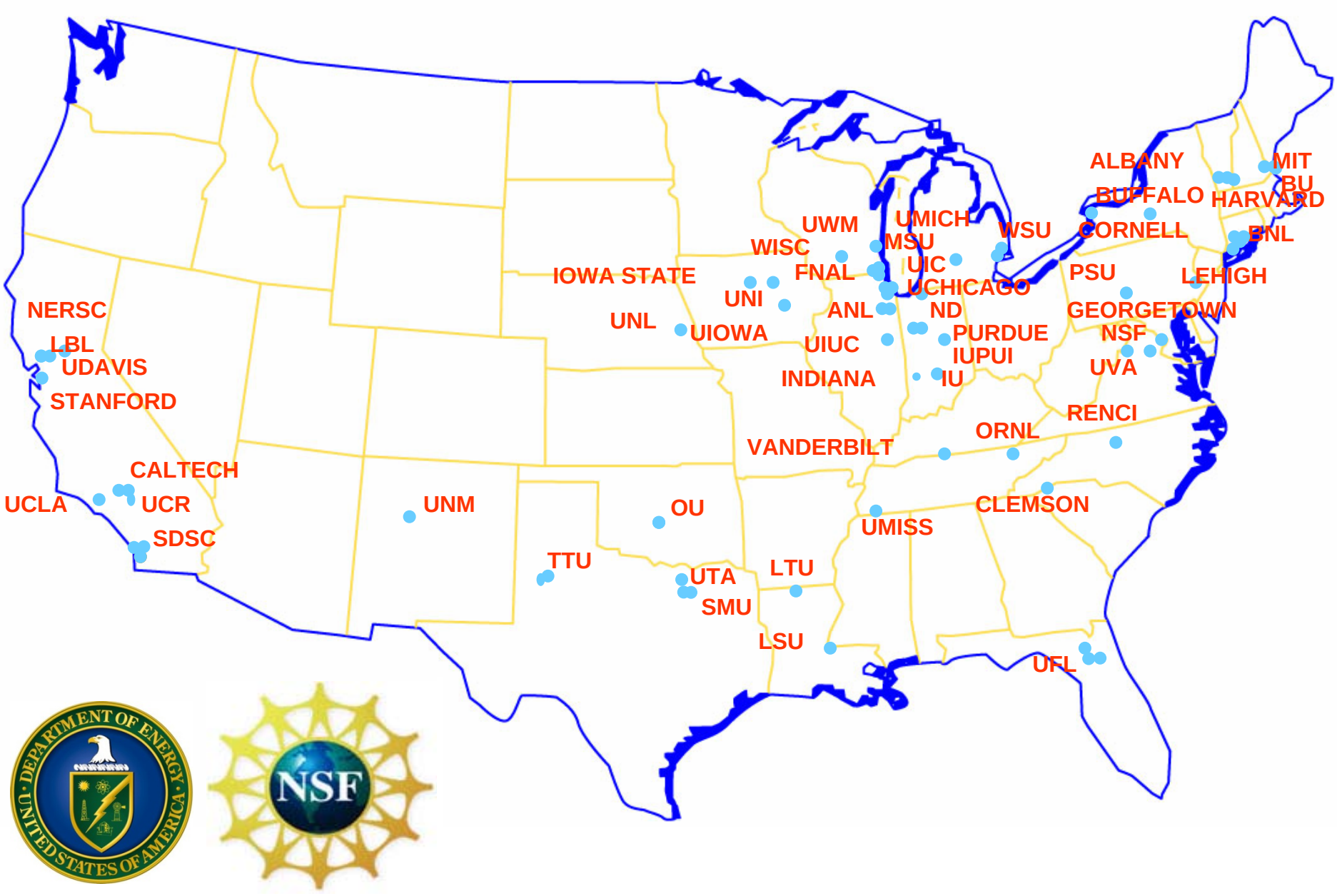
Distributed Resource Management-

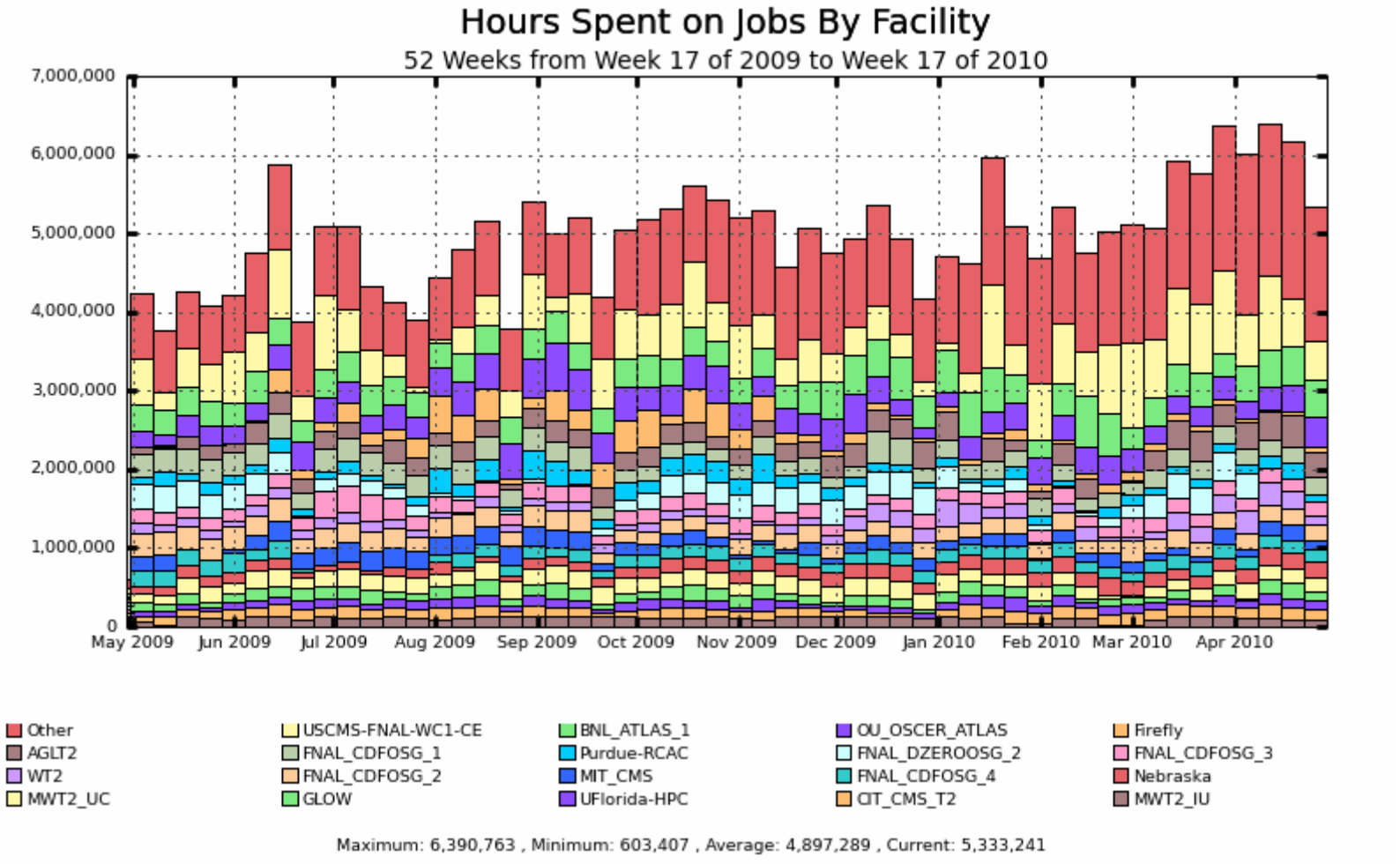
The Problem That Doesn't Go Away

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

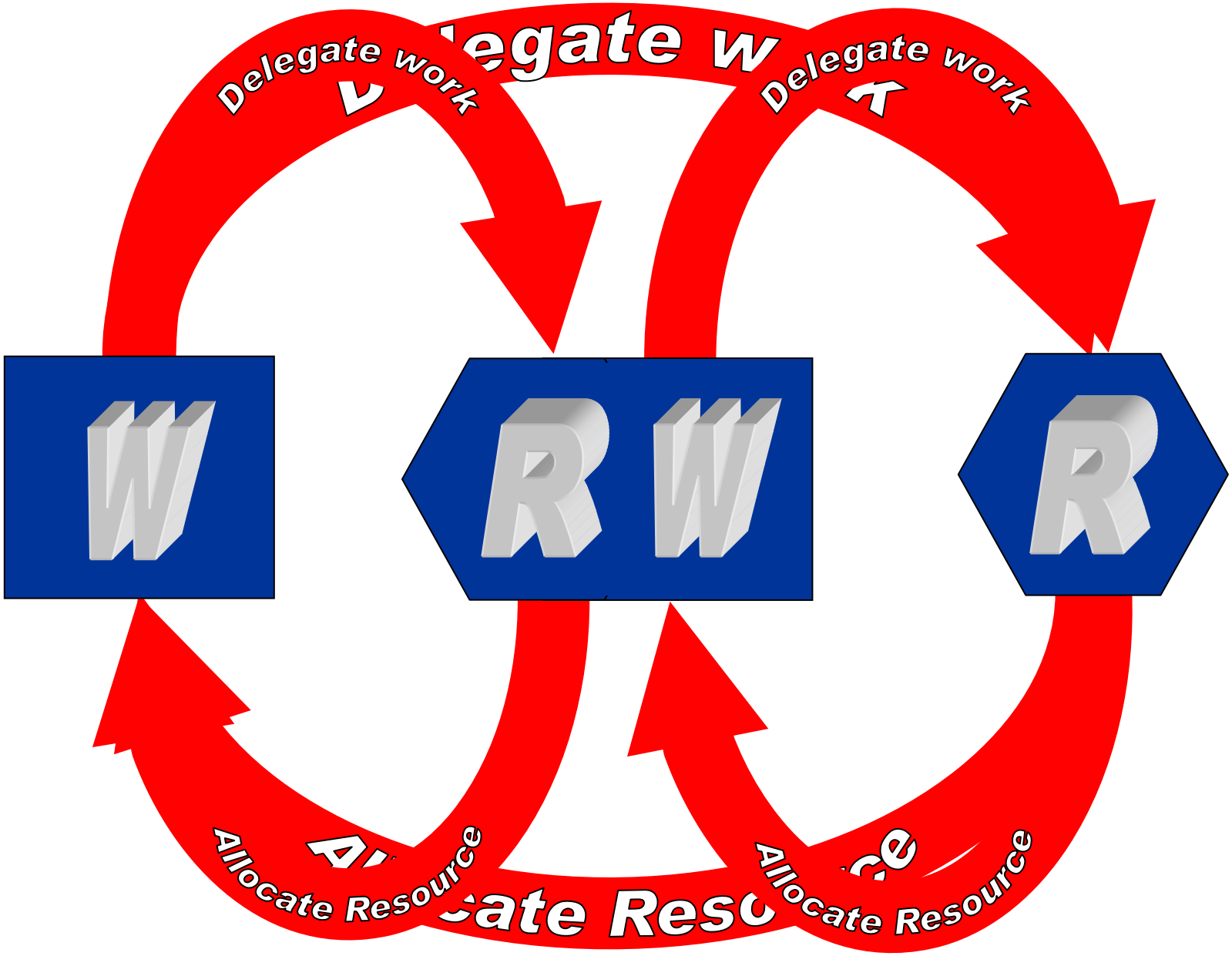
Open Science Grid (OSG)



OSG Today – CPU HOURS



Resource Allocation
(resource -> customer)
vs.
Work Delegation
(job/task -> resource)


Delegate work
Delegate work
W
R W
R
Allocate Resource
Allocate Resource

Resource Allocation

A limited assignment of temporary "ownership" of a resource offered by a provider to a requestor

- Requestor is charged for allocation regardless of actual consumption
- Requestor may be given the right to allocate resource to others
- Provider has the right and means to revoke the allocation
- Allocation is governed by an "agreement" between the provider and the requestor
- Allocation is a "lease" that expires if not renewed by the requestor
- Tree of allocations

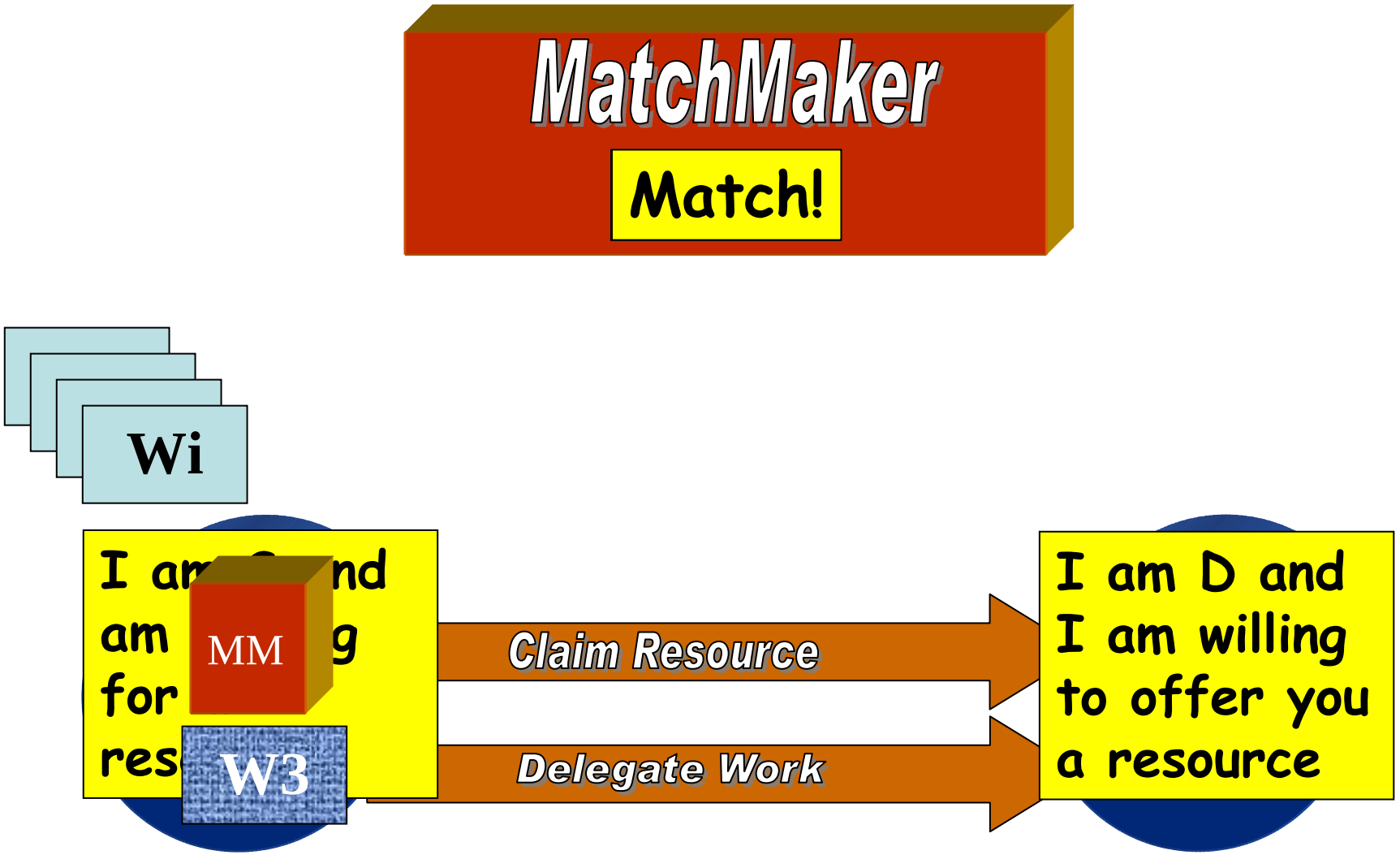
Work Delegation

An assignment of a responsibility to perform a task

- Delegation involved a definition of these "responsibilities"
- Responsibilities my be further delegated
- Delegation consumes resources
- Delegation is a "lease" that expires if not renewed by the assigner
- Can form a tree of delegations

Focus of the grid "movement" has been remote job delegation (Gate Keepers), commercial clouds are all about remote resource allocation (Provisioning)

In Condor we use a two phase matchmaking process to first allocate a resource to a requestor and then to select a task to be delegated to this resource


MatchMaker
Match!
Wi
I am
and
am
for
res
g
MM
W3
Claim Resource
Delegate Work
I am D and
I am willing
to offer you
a resource

Overlay Resource Managers

Ten years ago we introduced the concept of **Condor glide-ins** as a tool to support 'just in time scheduling' in a distributed computing infrastructure that consists of recourses that are managed by (heterogeneous) autonomous resource managers. By dynamically deploying a distributed resource manager on resources allocated (provisioned) by the local resource managers, the overlay resource manager can implement a unified resource allocation policy.

In other words, we use remote job invocation to get resources allocated.

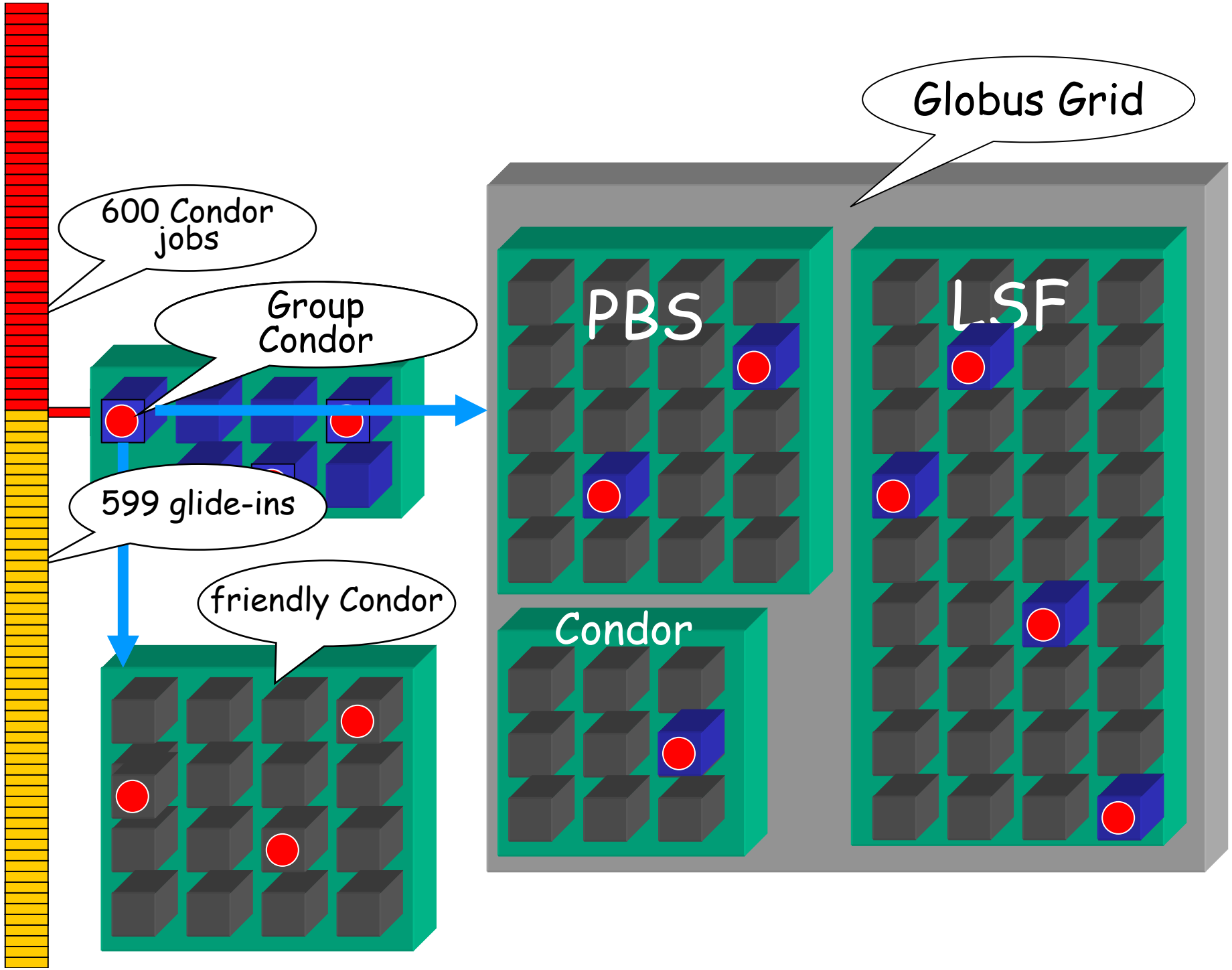
Slides from the First HEP Grid Workshop (at CHEP2000)

Step IV - Think big!

- Get access (account(s) + certificate(s)) to Globus managed Grid resources
- Submit 599 "To Globus" Condor glide-in jobs to your personal Condor
- When all your jobs are done, remove any pending glide-in jobs
- Take the rest of the afternoon off …

A "To-Globus" glide-in job will ...

- ... transform itself into a Globus job,
- submit itself to Globus managed Grid resource,
- be monitored by your personal Condor,
- once the Globus job is allocated a resource, it will use a GSIFTP server to fetch Condor agents, start them, and add the resource to your personal Condor,
- vacate the resource before it is revoked by the remote scheduler


Globus Grid
600 Condor jobs
Group Condor
599 glide-ins
friendly Condor
PBS
LSF
Condor

Today we have a very powerful and flexible 'GlideIn' framework that is used by the OSG (and cloud) community.

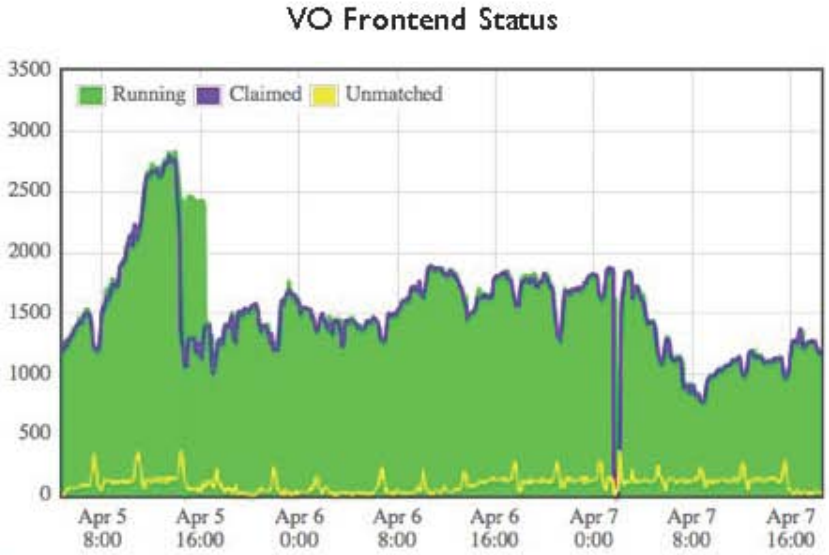
glideinWMS

Scalability achieved with 1 master collector and 70 slave collectors (on a single machine), machine with 16GB memory for hosting the schedd service:

Criteria	Design goal	Achieved so far
Total number of user jobs in the queue at any given time	100k	200k
Number of glideins in the system at any given time	10k	~26k
Number of running jobs per schedd at any given time	10k	~23k
Grid sites handled	~100	~100

glideinWMS: at Nebraska

- Running VO Frontend, Submitter, and Collector that uses the Factory at UCSD
- Used for combinatorics, biology, and bioinformatics applications
- Average 25,000 hours / day running jobs on glideinWMS
- 2.1 million CPU hours since beginning of 2010
- Currently limited by the memory of submit machine
  - Working on flocking local Condor Schedds to glideinWMS



Exposing a R-Allocation API

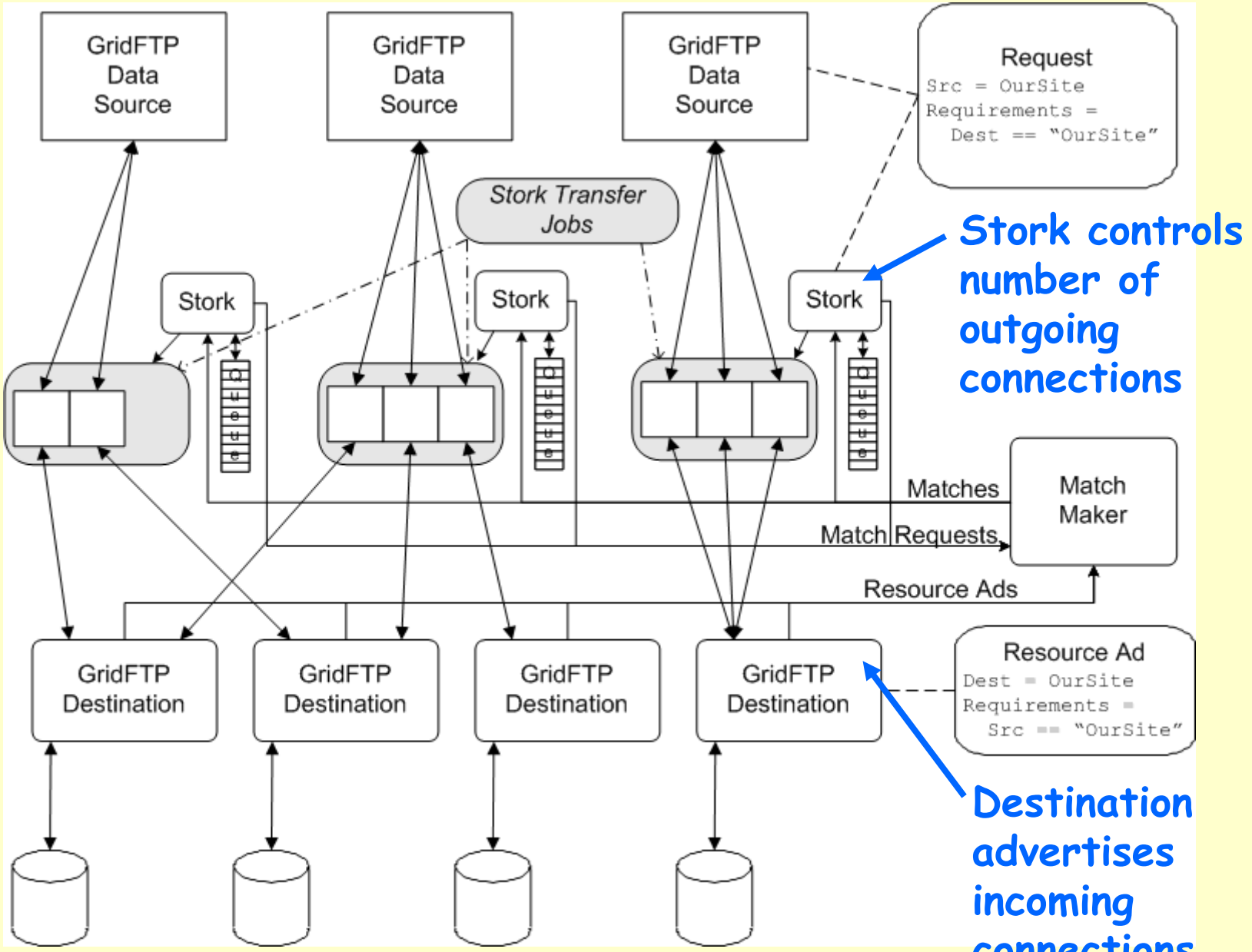
A growing number of groups (VOs) use Condor-G/Condor to provision resources. We are working on exposing the internal resource allocation primitives (claims) and adding (external) resource managers (give me up to 20, add 5, remove 3, …)

So far we have been talking (mainly) about CPUs/cores/slots/nodes. What about other resources like storage or data transfer capacity?

The SC'05 effort

Joint with the Globus GridFTP team

www.cs.wisc.edu/~miron


GridFTP Data Source
GridFTP Data Source
GridFTP Data Source
Request
Src = OurSite
Requirements =
Dest == "OurSite"
Stork Transfer Jobs
Stork
Stork
Stork
Stork controls number of outgoing connections
Matches
Match Maker
Match Requests
Resource Ads
GridFTP Destination
GridFTP Destination
GridFTP Destination
GridFTP Destination
Resource Ad
Dest = OurSite
Requirements =
Src == "OurSite"
Destination advertises incoming connections

WLCG has been using FTS

- Based on the abstraction of pre-defined channels with pre-defined allocation of resources per VO per channel
- Offers only partial control to endpoints
- Considered by the community as a "bandage"

From a 2003 Talk …

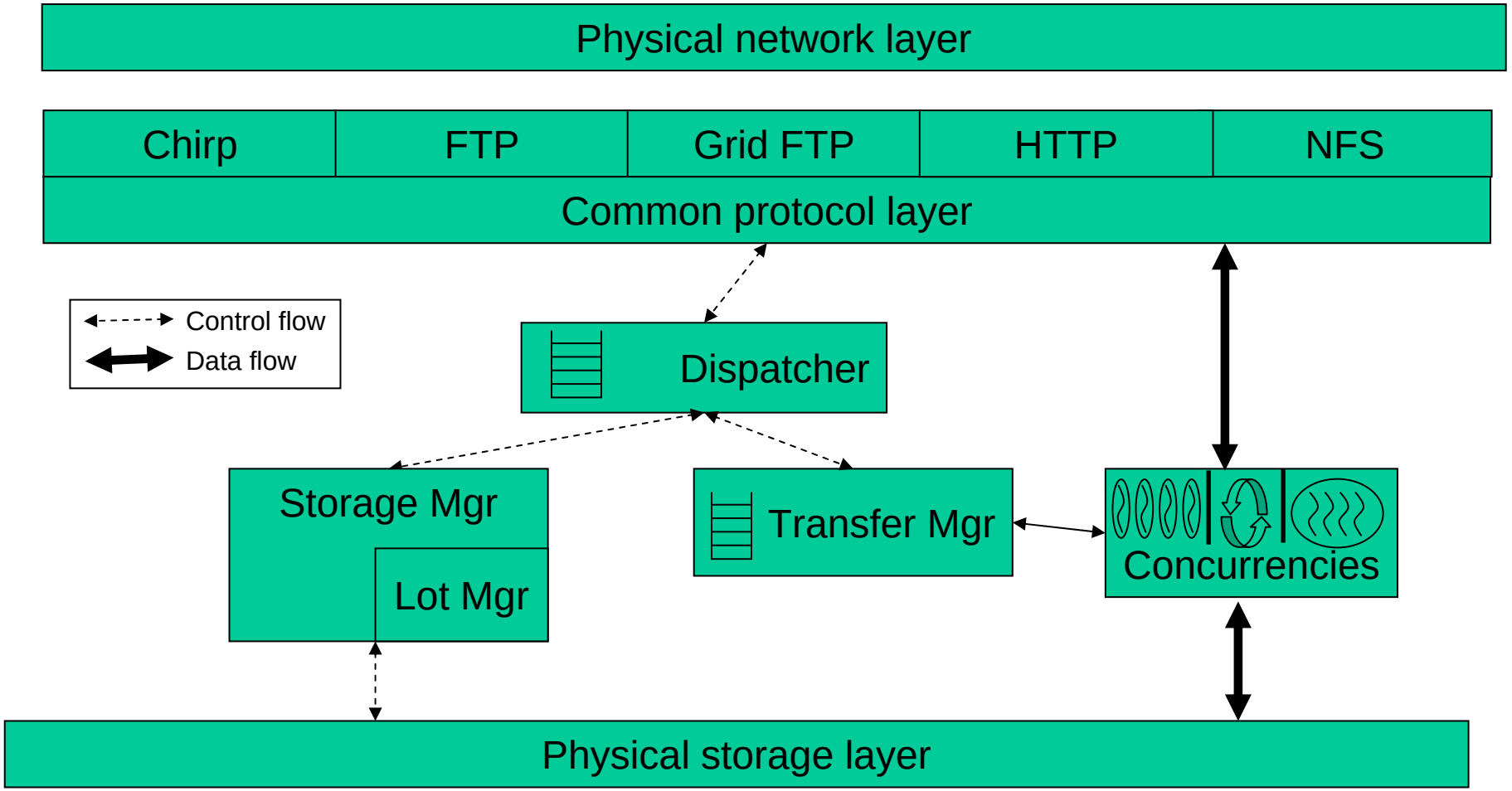
Overview of NeST

- Network storage server designed for the Grid
- Flexibility
  - Unprivileged, user-level software
  - Multiple protocols
    - Chirp, GridFTP, FTP, HTTP, NFS
  - Multiple concurrency models
    - Processes, threads, non-blocking
  - Portable

Overview of NeST

- Functionality
  - Exports Unix based file system
  - Space allocation (Lot)
- Manageability
  - User management – Dynamic users
  - Security – GSI authentication

NeST structure



Why space allocations ?

- Data generation and storage
  - Data intensive applications
  - Need to allocate space for temporary files
- Data migration
  - Many simultaneous data flows
    - Competition for storage space
    - Possible that all flows end in partial transfers
  - Need to ensure at least one succeeds

Space allocations in NeST

- Lot – abstraction for space allocation
- Create lot for a specified size and duration
- User and group lots
- Guaranteed and best-effort lots
- Hierarchical lots

Lot operations

- Create, Delete, Update
- MoveFile
  - Moves files across lots
- AddUser, RemoveUser
  - Lot level access control
  - List of users allowed to request sub-lots
- Attach / Detach
  - Associates a path to a lot

At this point only SRM provides 'some-sort' of space management capabilities. OSG uses it in a VERY limited way

Condor + HDFS

- Integrated approach to deploy (also on the fly) a distributed resource management system and a distributed file system
- Support asynchronous transfer of in/out sandbox of a job
- Add support for space management
- Offer caching across jobs
- Move files between Posix and HDFS interfaces

And what about energy provisioning?

From Forbes Magazine …

Open Source Energy Savings

Dan Woods, 03.02.10

Software for spreading work over huge collections of computers can be used to cut power costs.

Condor supports all the operating systems a typical company or research institution would have and is **rock solid** in terms of stability and functions for its intended purpose, which is carving up work and sending it out to any number of computers for processing.

TIME *A thinker's guide to the most important trends of the new decade*

In Defense of Failure

By Megan McArdle Thursday, Mar. 11, 2010

"The goal shouldn't be to eliminate failure; it should be to build a system resilient enough to withstand it"

"The real secret of our success is that we learn from the past, and then we forget it. Unfortunately, we're dangerously close to forgetting the most important lessons of our own history: how to fail gracefully and how to get back on our feet with equal grace."