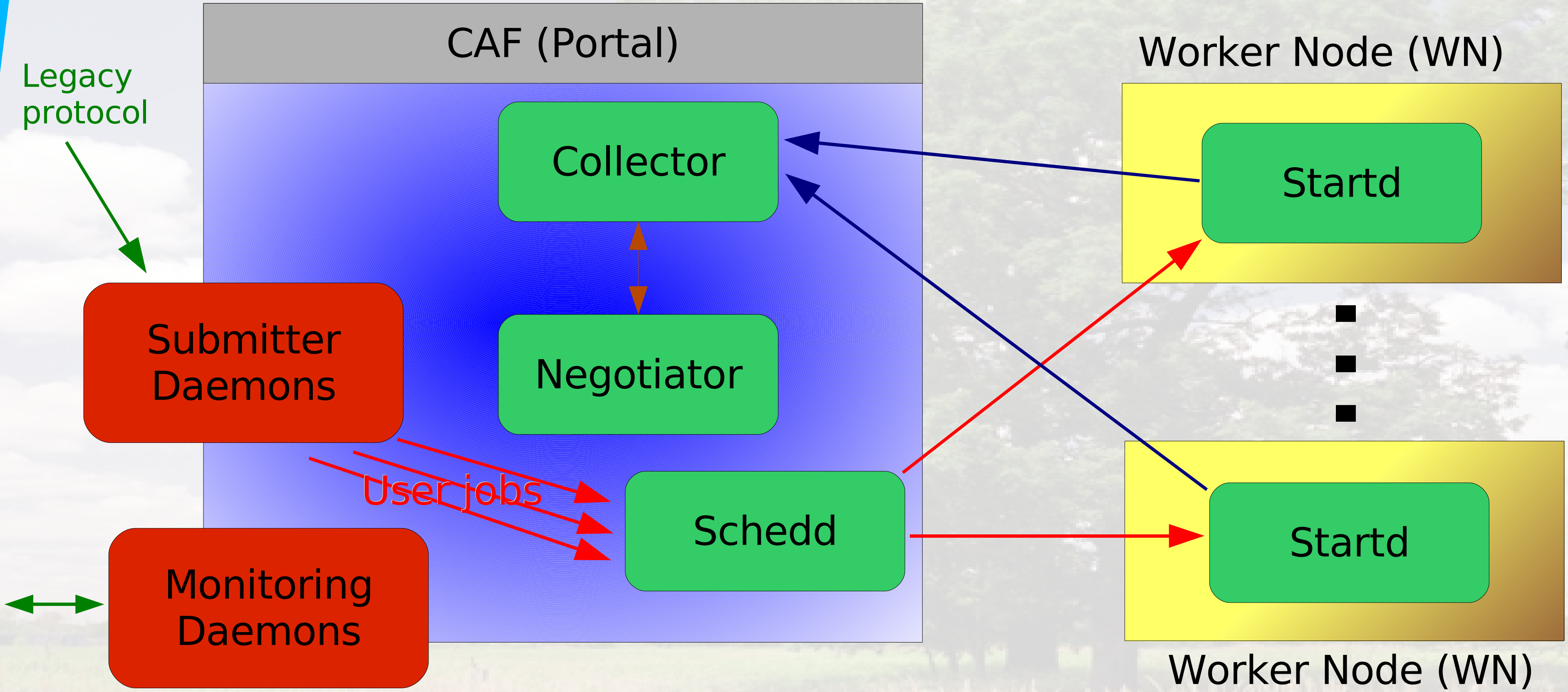
Using Condor Glide-ins and GCB to run in a grid environment

Elliot Lipeles, Matthew Norman, Frank Würthwein,
University of California, San Diego

Subir Sarkar, Igor Sfiligoi,
INFN

Traditional CDF use of Condor

CDF Analysis Farms (CAF)

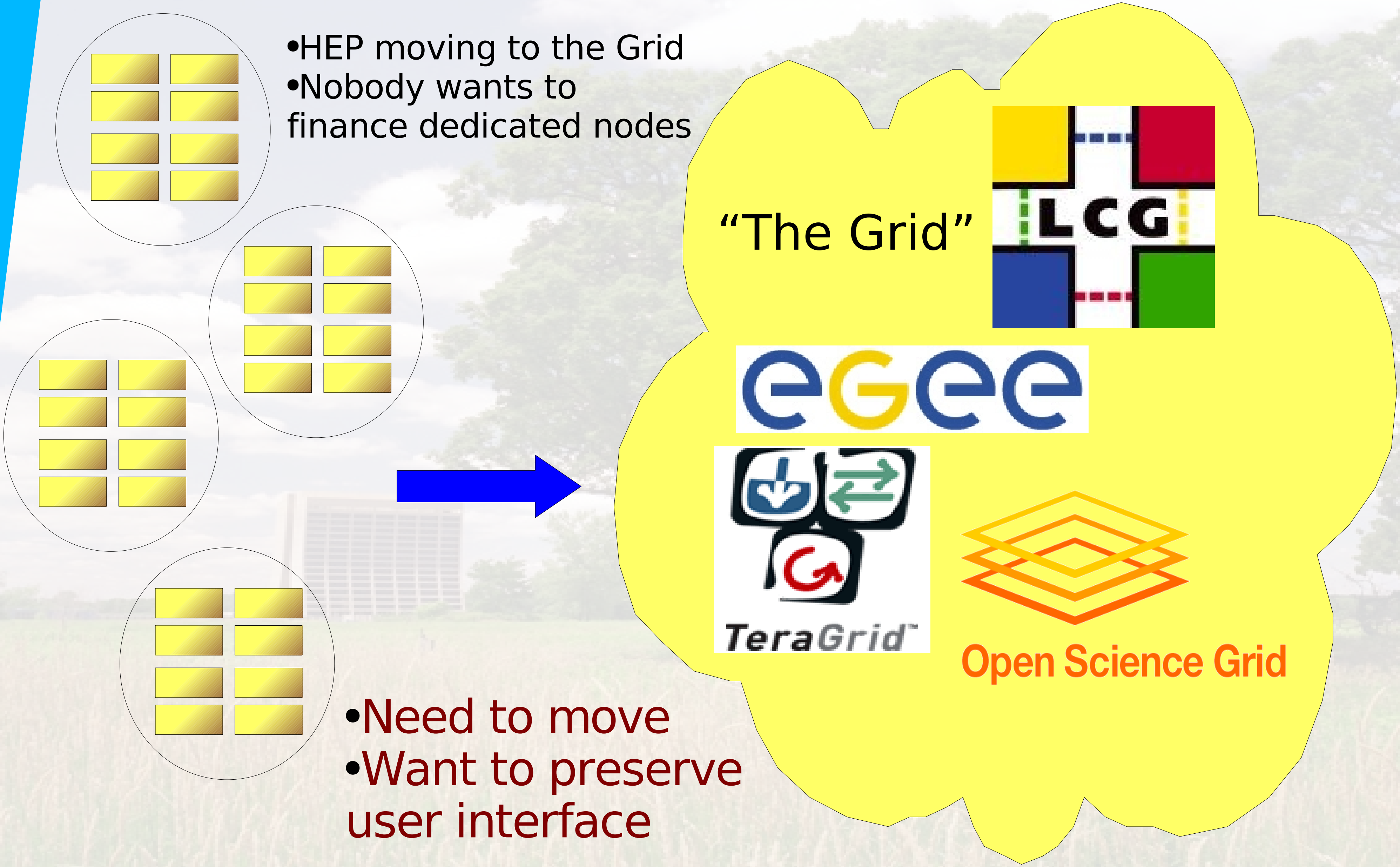


Single Point of Submission system

- CAF Daemons accept and authenticate user jobs, handle output.
- Condor Schedd does the real job

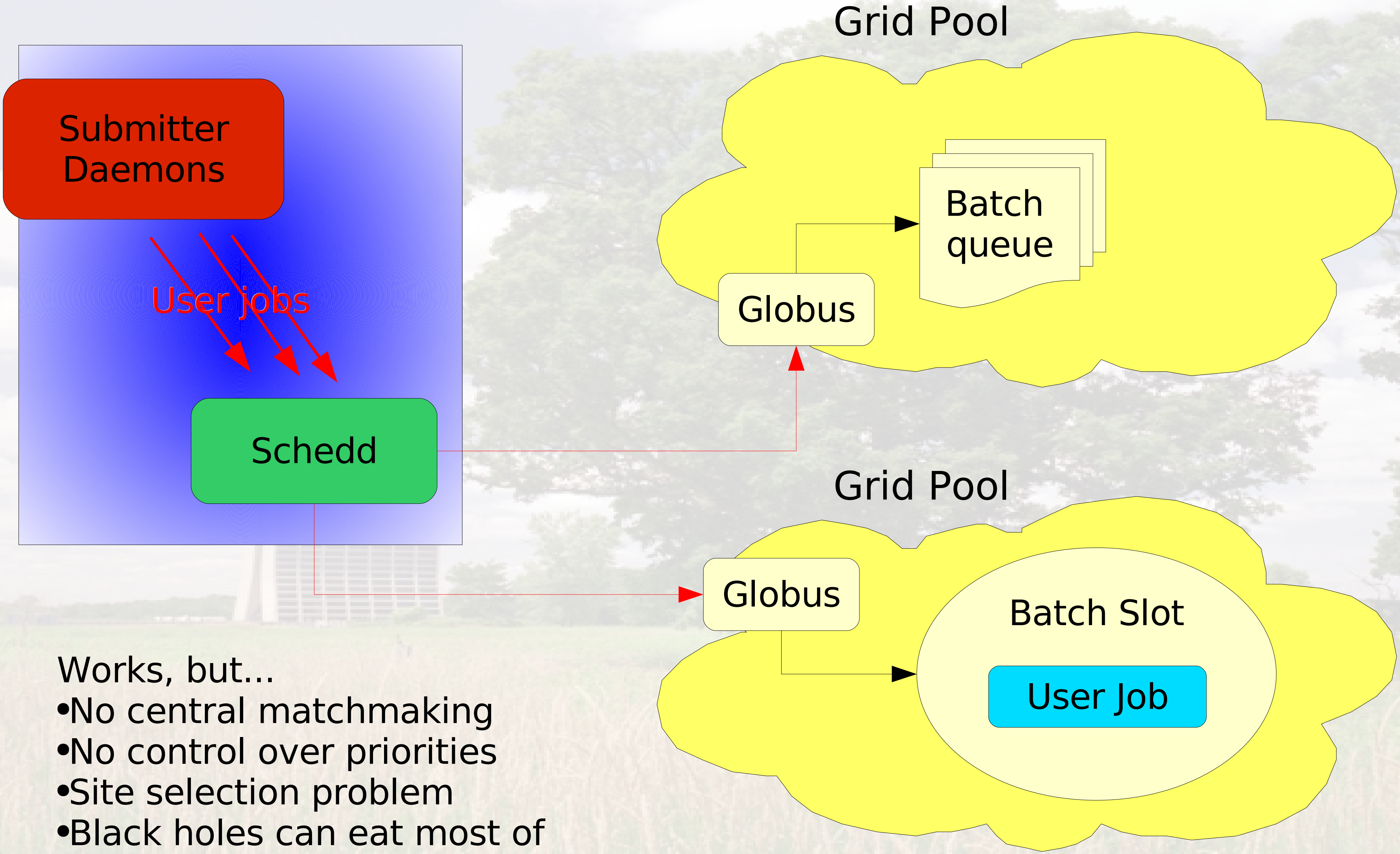
Need to move to the Grid

- HEP moving to the Grid
- Nobody wants to finance dedicated nodes



- Need to move
- Want to preserve user interface

Plain Condor-G?



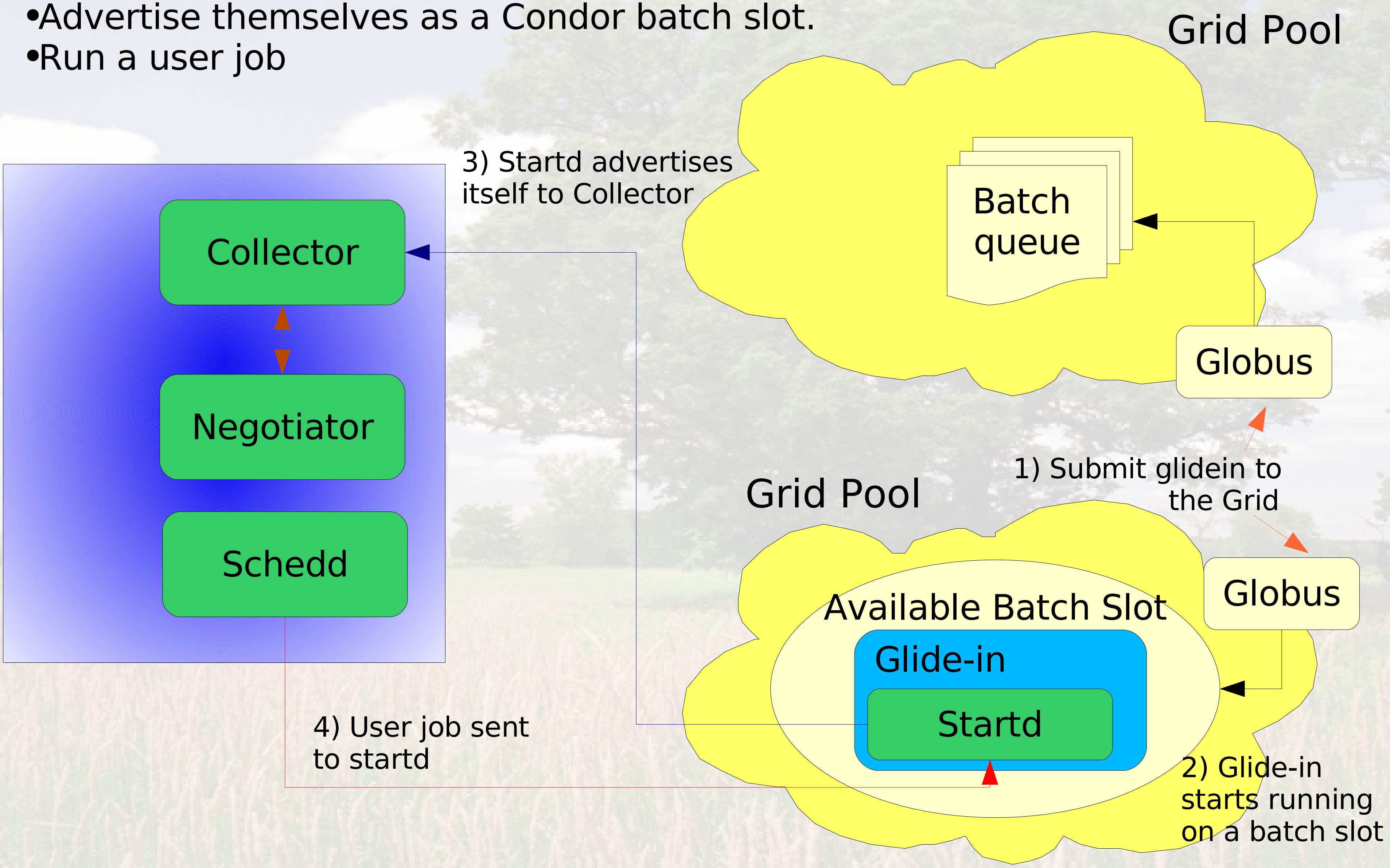
Works, but...

- No central matchmaking
- No control over priorities
- Site selection problem
- Black holes can eat most of the user jobs

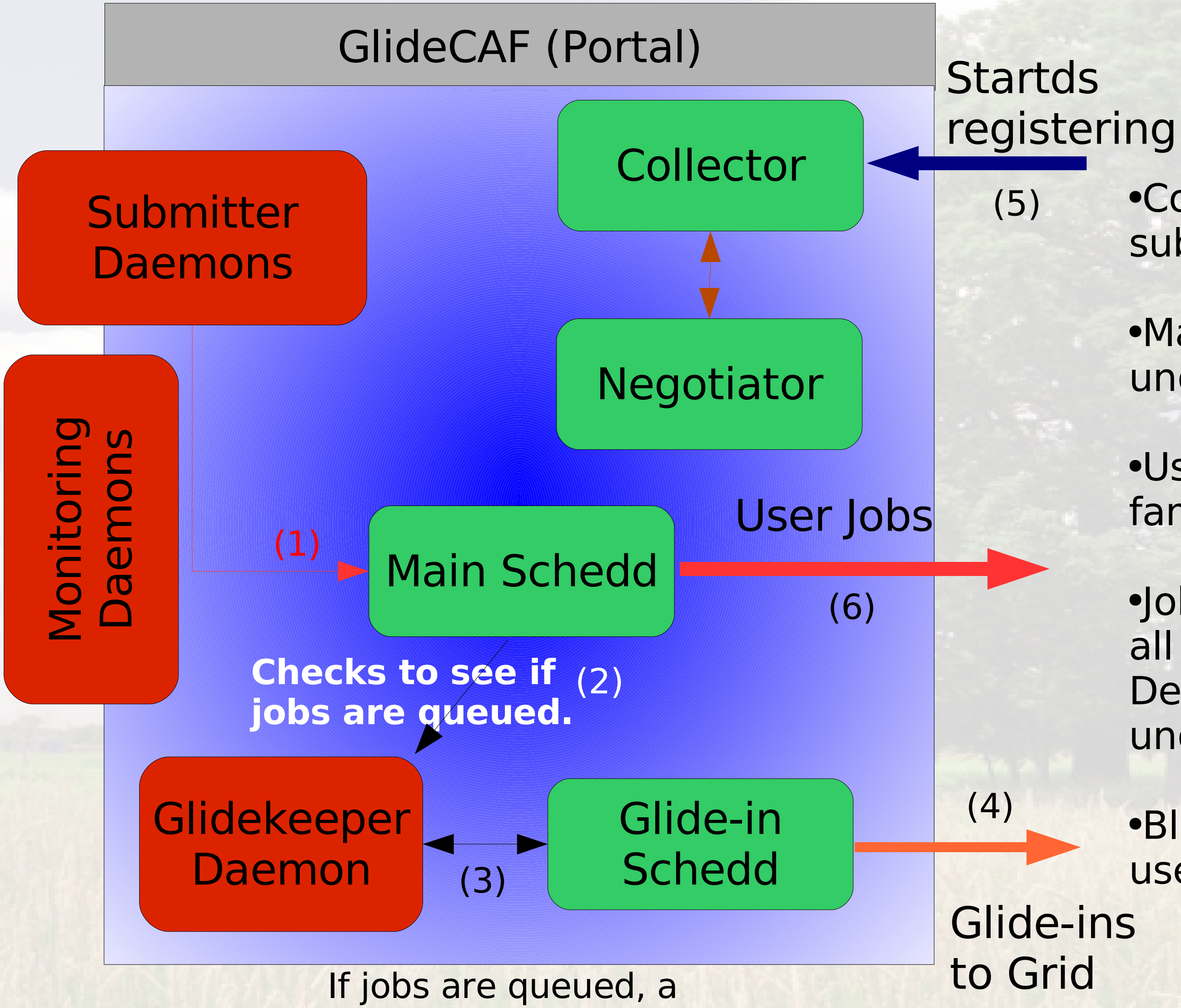
Condor Glide-Ins

Turn shared Grid resources into temporary private condor resources:

- Submit as a standard Grid job.
- Advertise themselves as a Condor batch slot.
- Run a user job


Grid Pool
3) Startd advertises itself to Collector
Batch queue
Collector
Negotiator
Schedd
Globus
1) Submit glidein to the Grid
Grid Pool
Globus
Available Batch Slot
Glide-in
Startd
4) User job sent to startd
2) Glide-in starts running on a batch slot

CDF use of Glide-Ins



If jobs are queued, a glide-in is submitted to a second schedd.

- •Condor allows single point of submission to Grid sites.
- •Matchmaking done globally under CDF control.
- •Users are presented with a familiar interface.
- •Jobs run as Condor jobs with all features (such as Condor on Demand) regardless of underlying site batch system.
- •Black holes eat glide-ins, not user jobs

Glide-Ins are just regular Startds

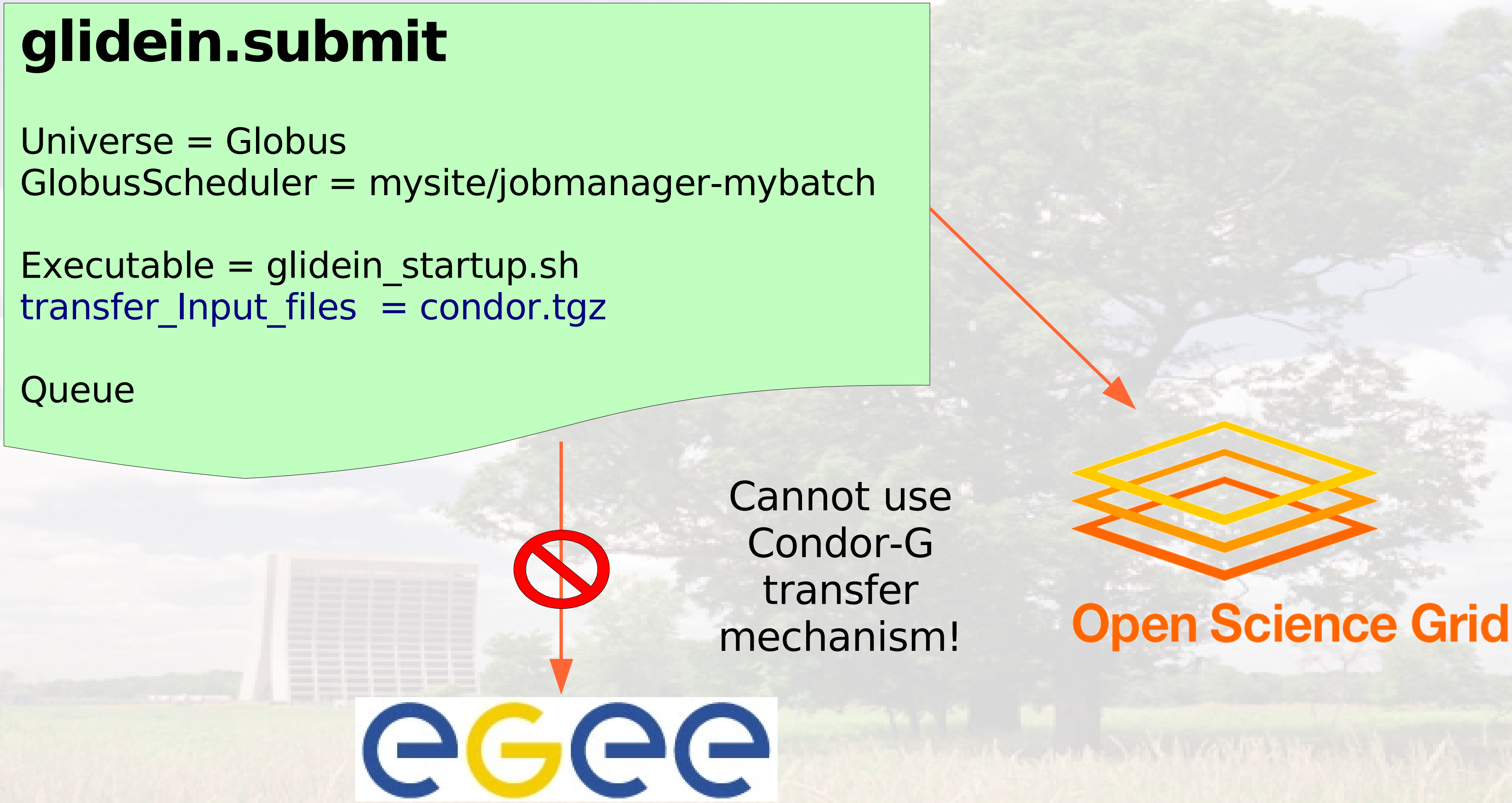
condor_config

```
DAEMON_LIST = MASTER,STARTD
NEGOTIATOR_HOST = $(HEAD_NODE)
COLLECTOR_HOST = $(HEAD_NODE)

MaxJobRetirementTime=$(SHUTDOWN_GRACEFUL_TIMEOUT)
SHUTDOWN_GRACEFUL_TIMEOUT=288000

# How long will it wait in an unclaimed state before exiting
STARTD_NOCLAIM_SHUTDOWN = 1200

HEAD_NODE = cdfhead.fnal.gov
```

glidein_startup.sh

```
validate_node()
local_config()
./condor_master -r $retmins -dyn -f
```

Delivering files with Condor-G



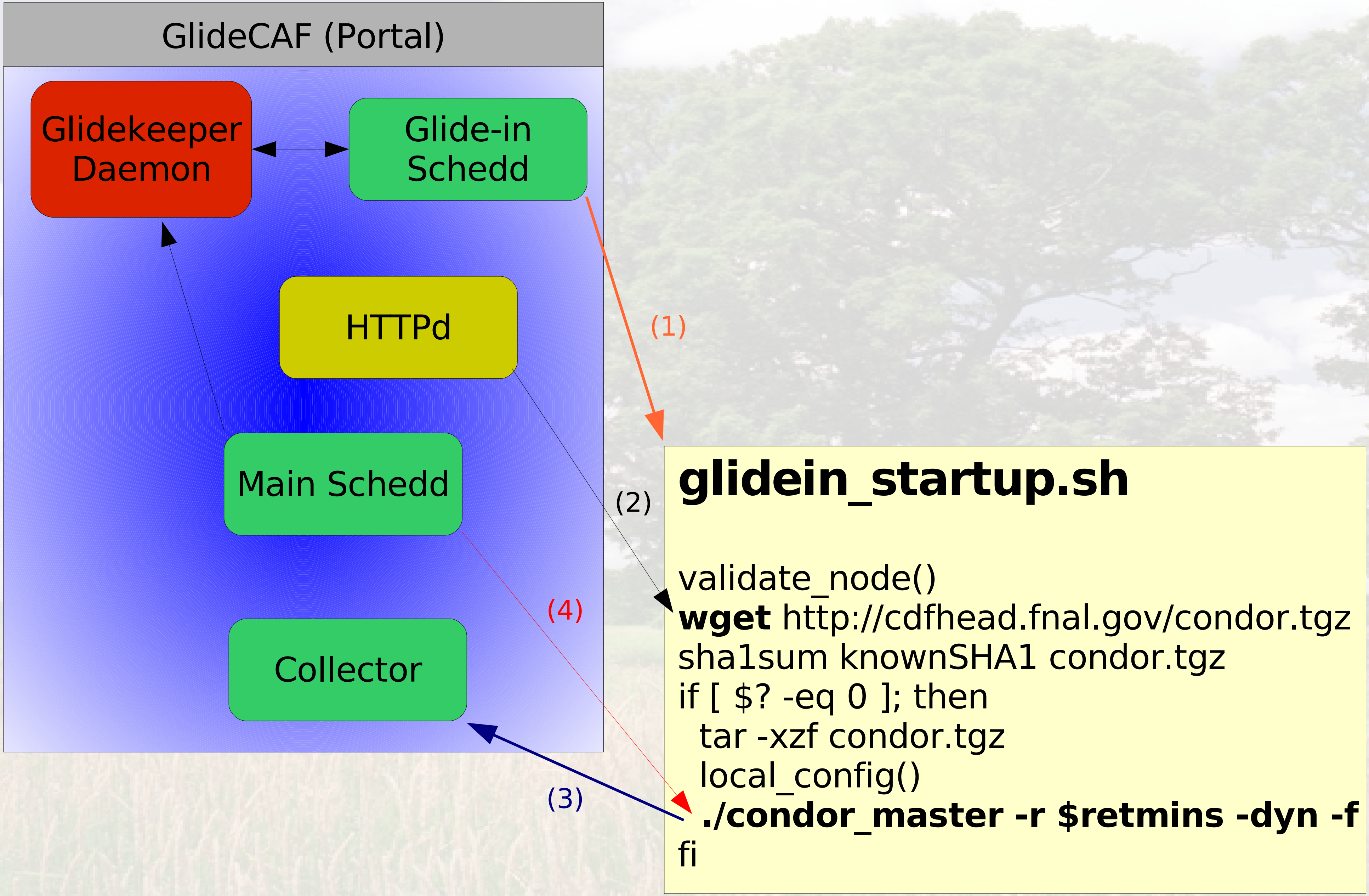
glidein.submit

```
Universe = Globus
GlobusScheduler = mysite/jobmanager-mybatch

Executable = glidein_startup.sh
transfer_Input_files  = condor.tgz

Queue
```

Cannot use Condor-G transfer mechanism!

Only the executable guaranteed to be transfered on EGEE.

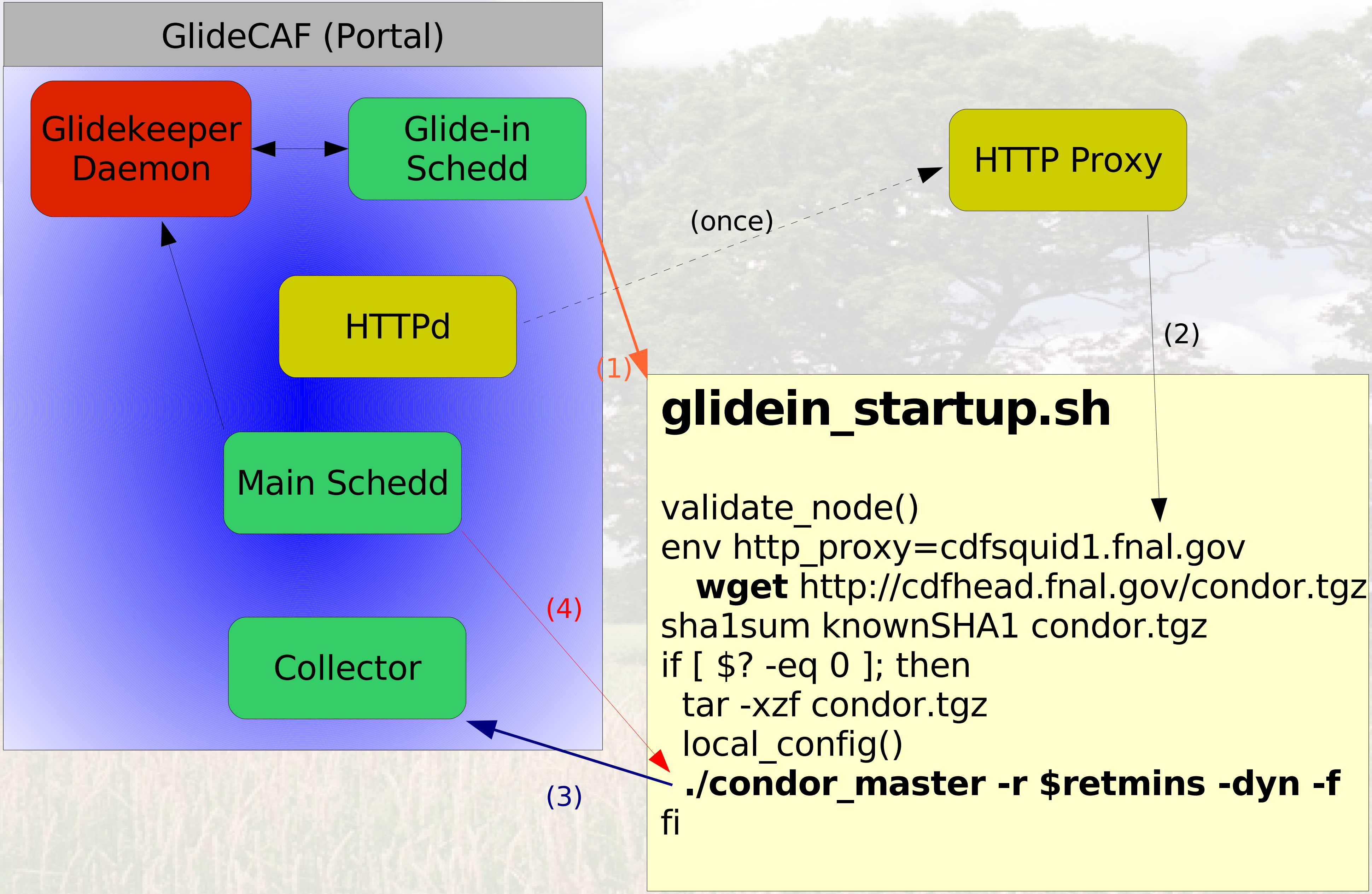
Use a Web server for file delivery



glidein_startup.sh

```
validate_node()
wget http://cdfhead.fnal.gov/condor.tgz
sha1sum knownSHA1 condor.tgz
if [ $? -eq 0 ]; then
  tar -xzf condor.tgz
  local_config()
  ./condor_master -r $retmins -dyn -f
fi
```

HTTP Proxy can reduce traffic to head node

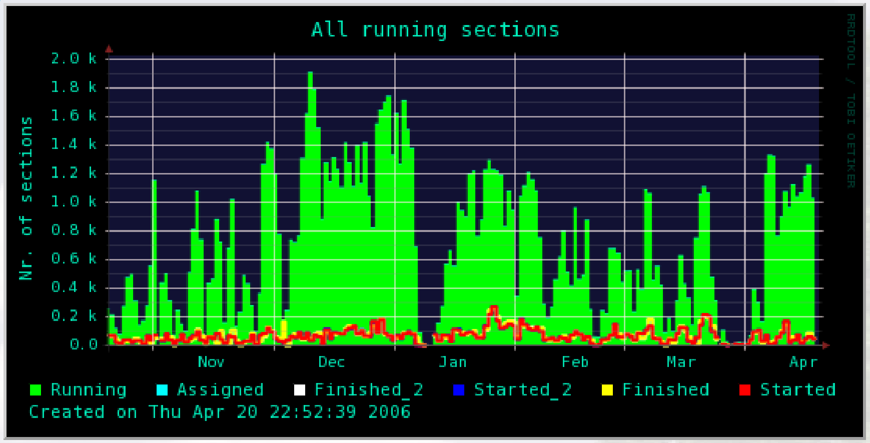


glidein_startup.sh

```
validate_node()
env http_proxy=cdfsquid1.fnal.gov
   wget http://cdfhead.fnal.gov/condor.tgz
sha1sum knownSHA1 condor.tgz
if [ $? -eq 0 ]; then
  tar -xzf condor.tgz
  local_config()
  ./condor_master -r $retmins -dyn -f
fi
```

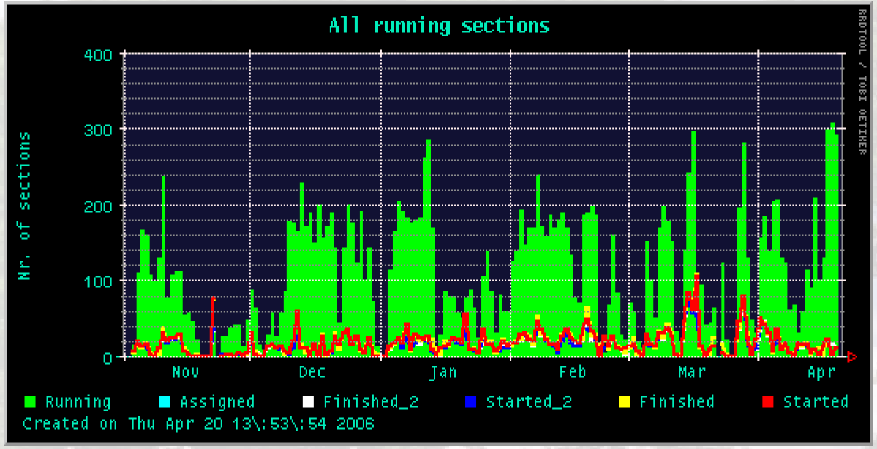
Several GlideCAFs deployed

CNAF (Bologna, Italy)



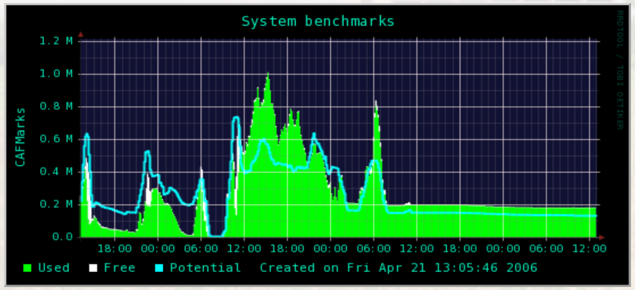
CNAF six month history plot: Note over a thousand VMs used for long period of time.

SDSC (San Diego, CA, USA)

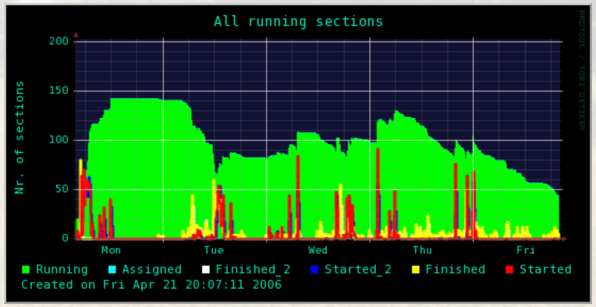


SDSC History Plot showing number of VMs run.

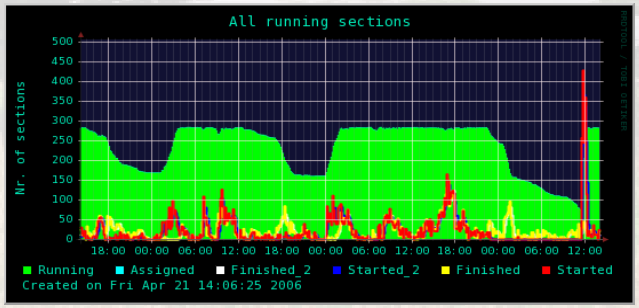
Fermilab (Batavia, IL, USA)



IN2P3 (Lyon, France)

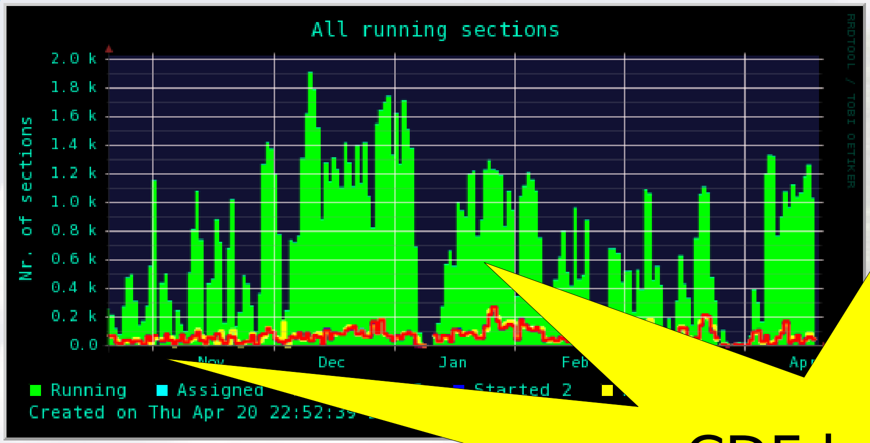


MIT (Boston, MA, USA)

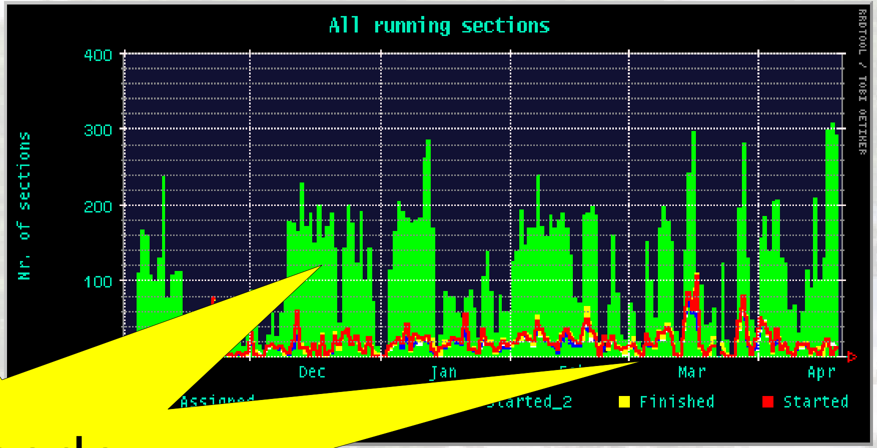


Several GlideCAFs deployed

CNAF (Bologna, Italy)



SDSC (San Diego, CA, USA)



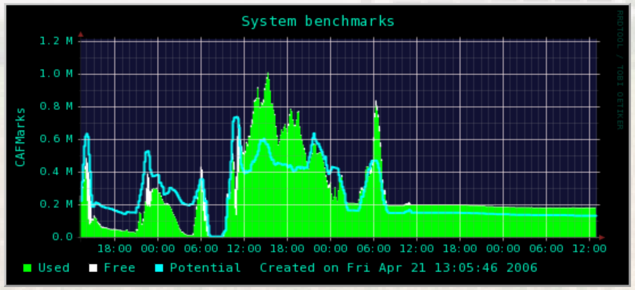
CNAF six month hi

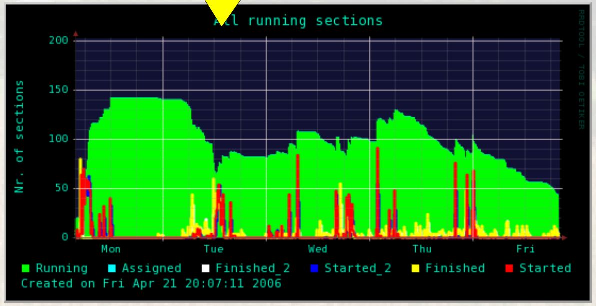
thousand VMs used for

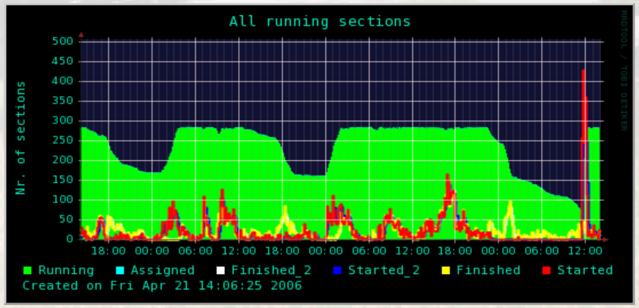
CDF head node
near the worker nodes

ry Plot showing number of

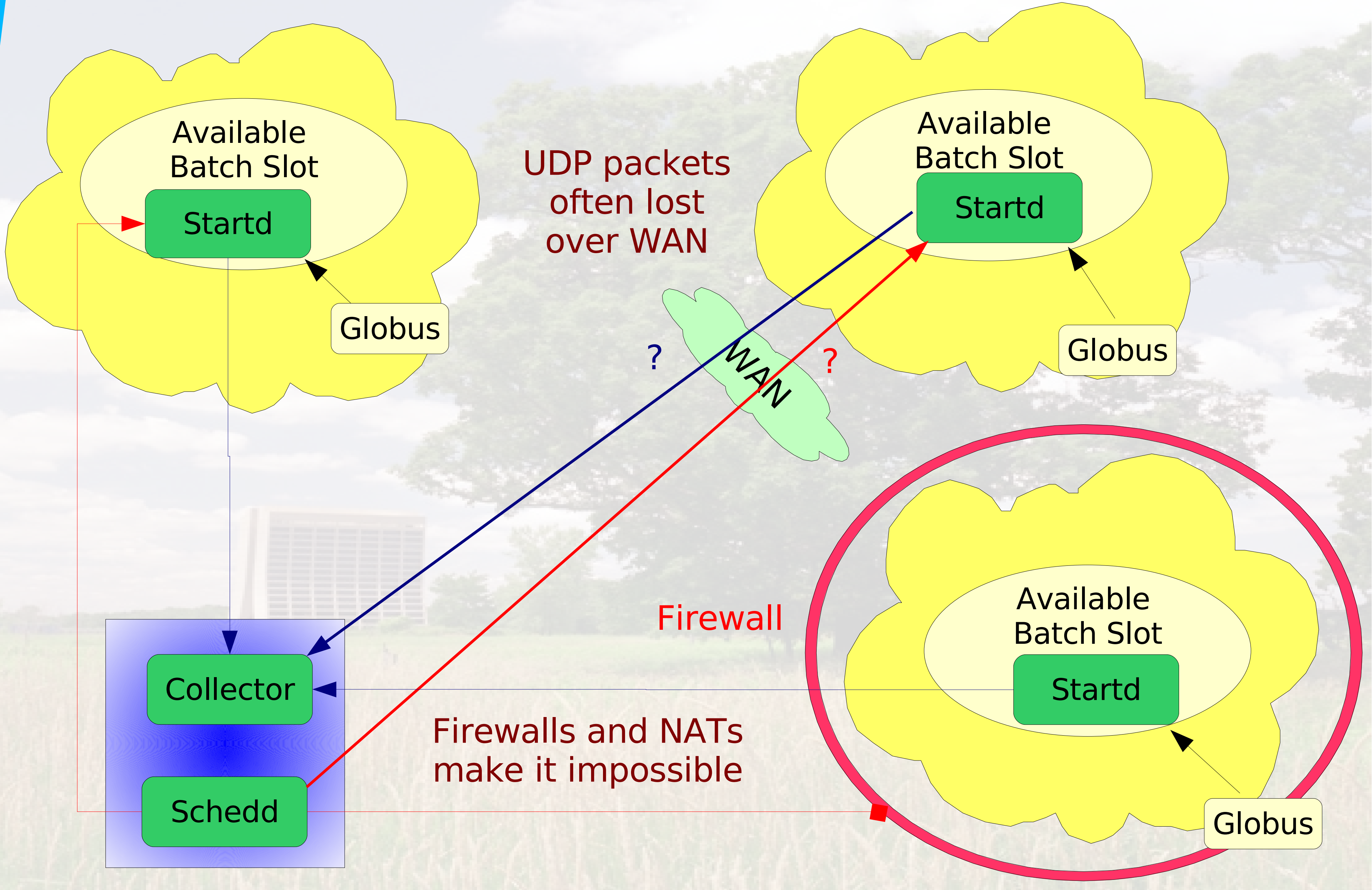
Fermilab (Batavia, IL, USA) IN2P3 (Lyon, France) MIT (Boston, MA, USA)



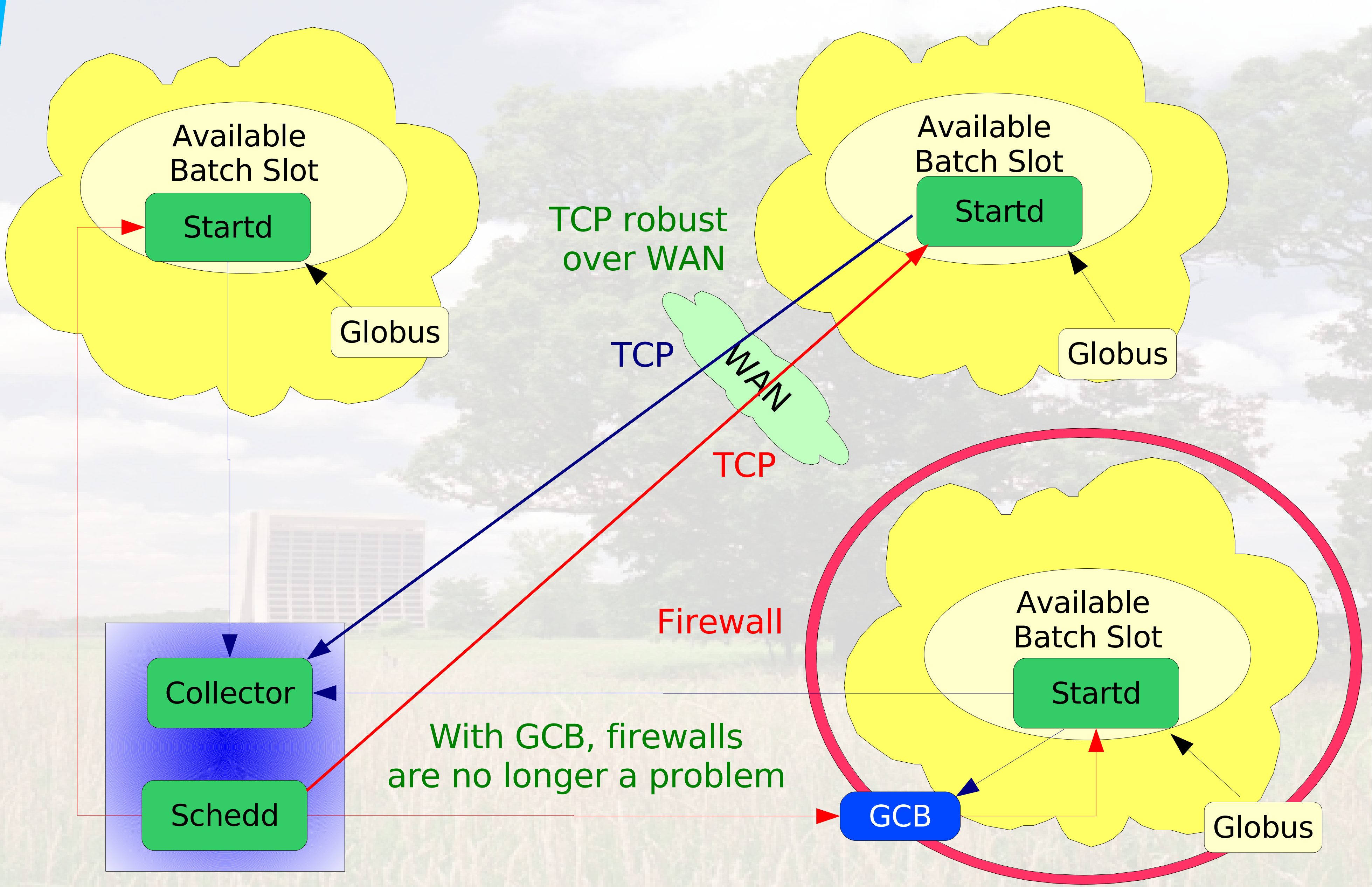




Gliding to multiple sites a problem



The Solution? GCB and TCP



With GCB, firewalls are no longer a problem

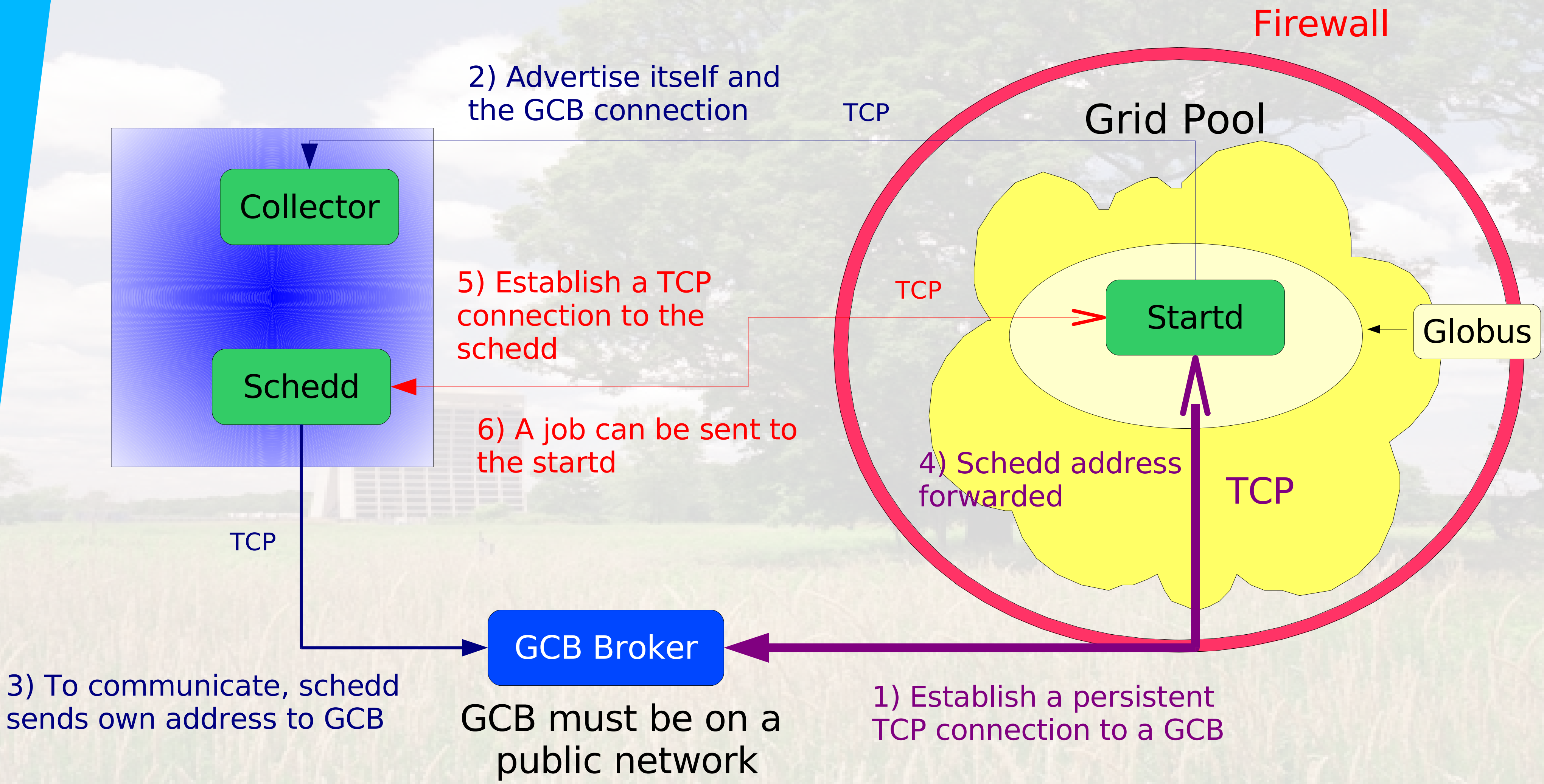
Generic Connection Brokering(GCB)

GCB

•Now shipped with current Condor development versions

•Designed to allow connections to pass over firewalls

•Part of Condor – No software to install on Grid site


Firewall
2) Advertise itself and the GCB connection
TCP
Grid Pool
Collector
5) Establish a TCP connection to the schedd
TCP
Startd
Globus
Schedd
6) A job can be sent to the startd
4) Schedd address forwarded
TCP
TCP
GCB Broker
3) To communicate, schedd sends own address to GCB
GCB must be on a public network
1) Establish a persistent TCP connection to a GCB

Enabling GCB just a matter of configuration

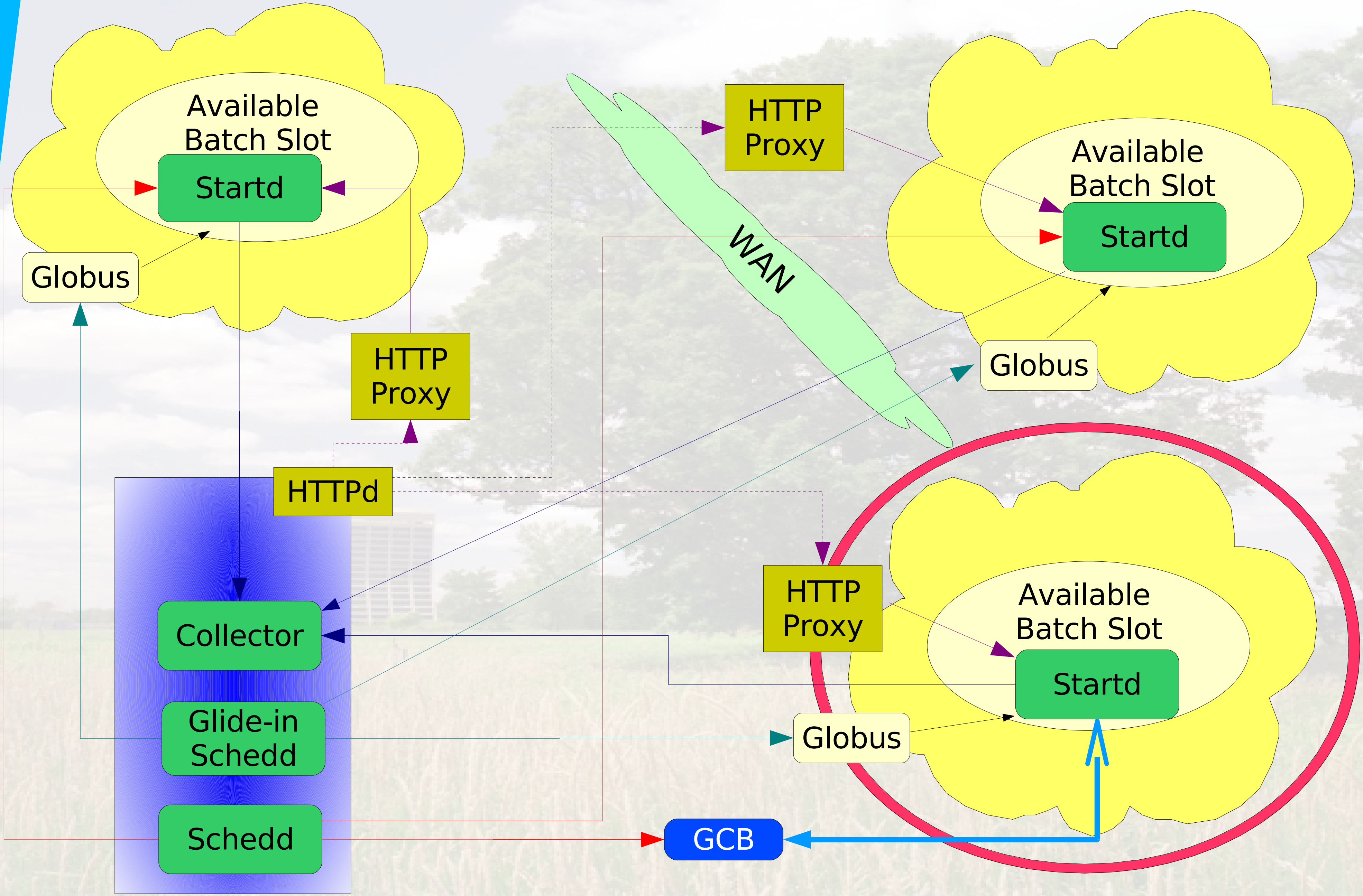
condor_config

```
DAEMON_LIST = MASTER,STARTD
NEGOTIATOR_HOST = $(HEAD_NODE)
COLLECTOR_HOST = $(HEAD_NODE)

MaxJobRetirementTime=$(SHUTDOWN_GRACEFUL_TIMEOUT)
SHUTDOWN_GRACEFUL_TIMEOUT=288000

# How long will it wait in an unclaimed state before exiting
STARTD_NOCLAIM_SHUTDOWN = 1200

HEAD_NODE = cdfhead.fnal.gov

# GCB configuration
NET_REMAP_ENABLE = true
NET_REMAP_SERVICE = GCB
NET_REMAP_INAGENT = cdfgcb1.fnal.gov
```

tup.sh

```
ad.fnal.gov/condor.tgz
HA1 condor.tgz
n
  -xzf condor.tgz
  local_config()
  ./condor_master -r $retmins -dyn -f
fi
```

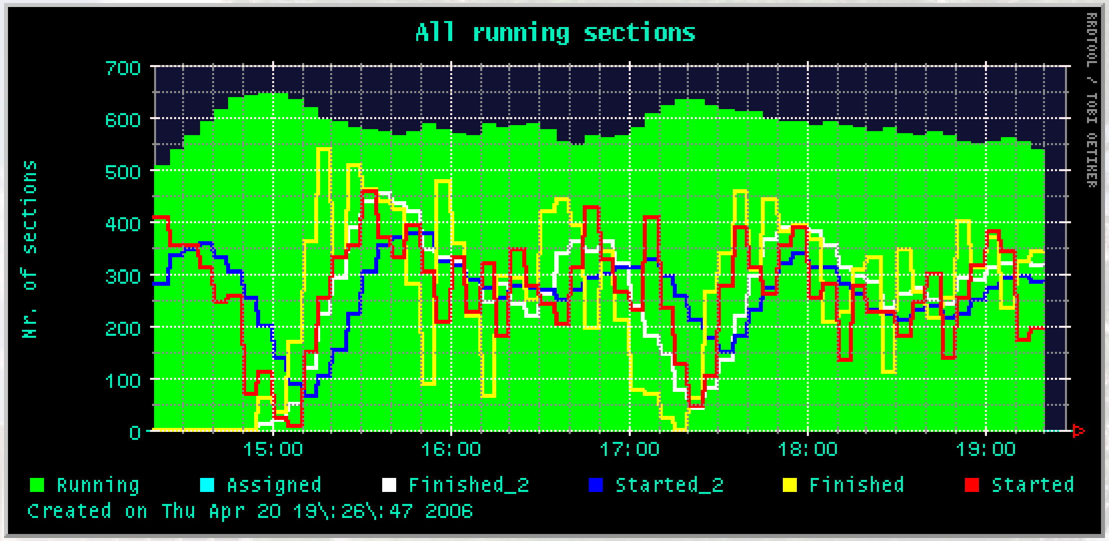
CDF on the Grid



Status of the GCB-based Condor pool

GCB Performance is currently being tested:

- Can coordinate up to 400-500 VMs from a single modified GCB node
- Functions well at all sites that allow outbound connectivity



Scaling test : ~600 VMs running on multiple grid sites

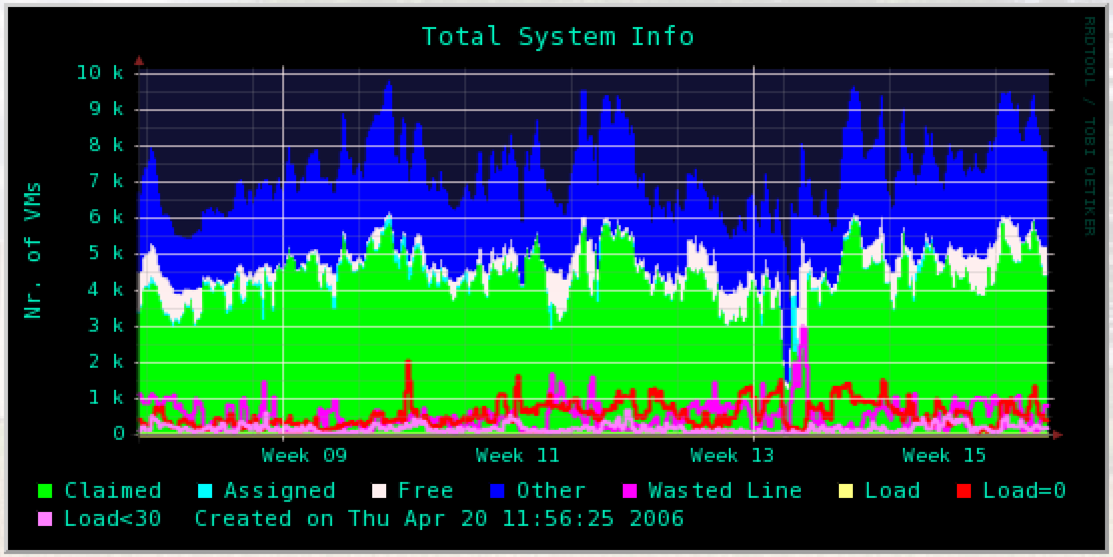
Future plans

- Latest official version had scaling problems: we are testing a pre-release
- Attempts will be made to scale past several hundred VMs

Future Development

We would like to use condor to combine our current clusters into one super-cluster.

This means testing to confirm that Condor scales to the appropriate level.



Current sum total resources

Summary

Using Condor took CDF to the Grid

- Switching to glide-ins was easy
- Matchmaking done globally under CDF control
  - Late binding removes the necessity to choose between sites
- No need for any add-ons at the Grid site
- We could preserve our interface
- We look forward to working with Condor in the future