

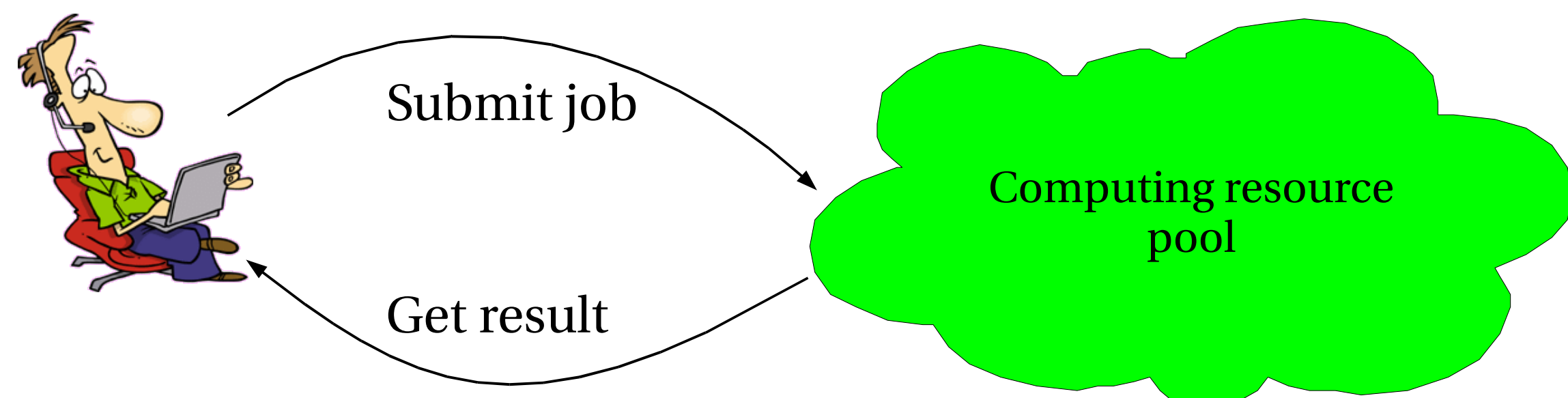
Interoperability and Scalability within glideinWMS

D. Bradley¹, I. Sfiligoi², S. Padhi³, J. Frey¹, T. Tannenbaum¹

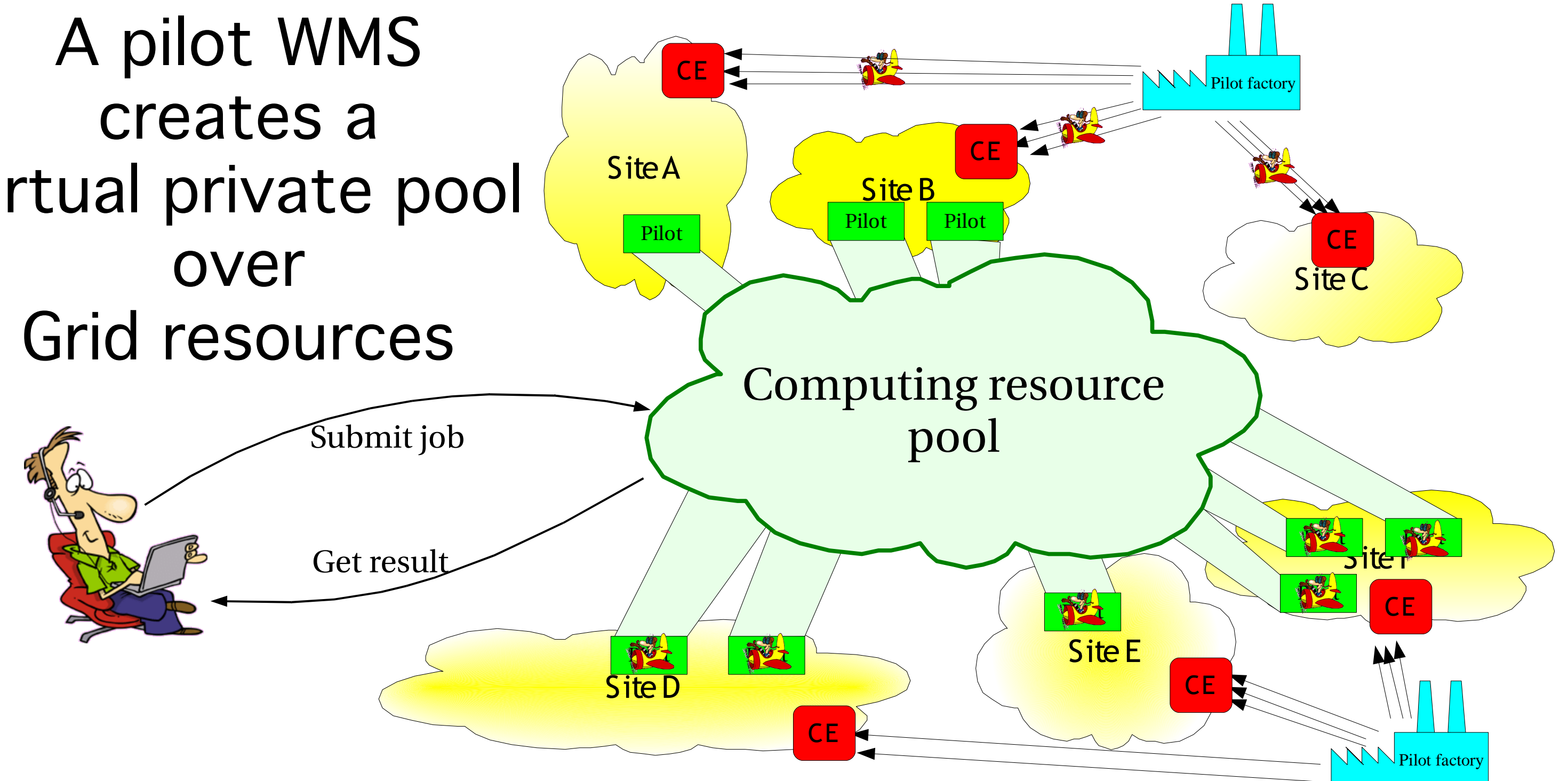
¹University of Wisconsin, ²Fermilab, ³University of Southern California San Diego

First the Basics: why use glideinWMS?

Users want simple access to compute resources

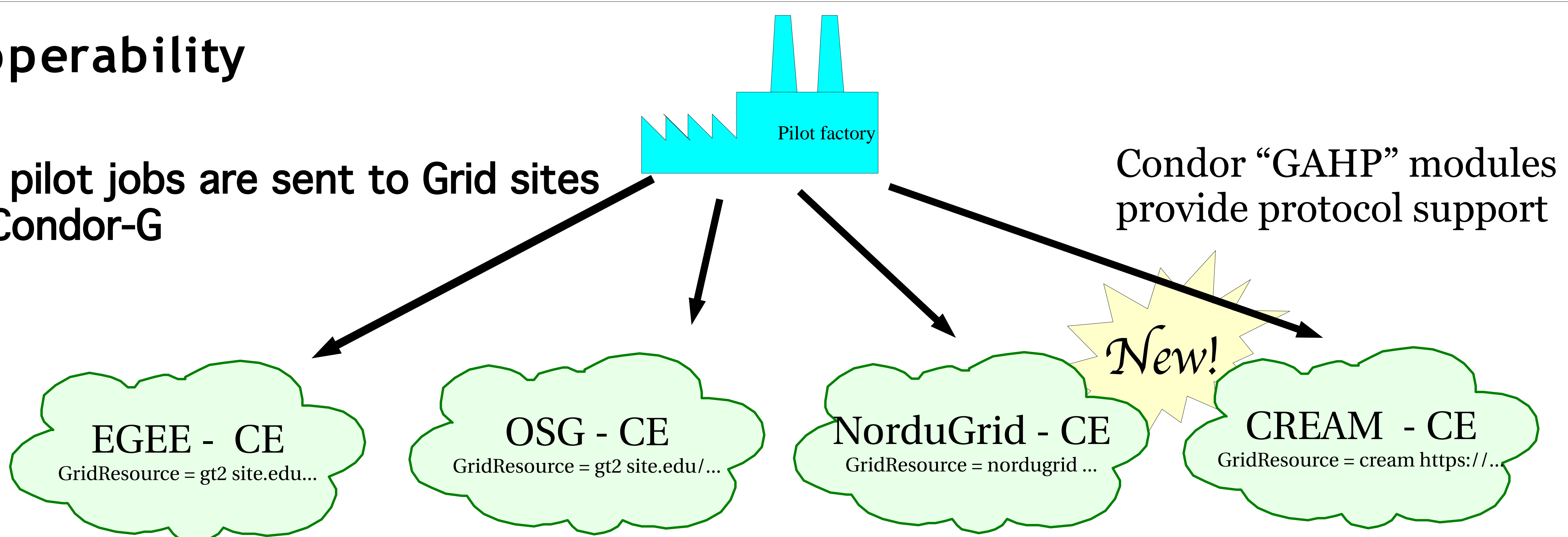


A pilot WMS creates a virtual private pool over Grid resources



Interoperability

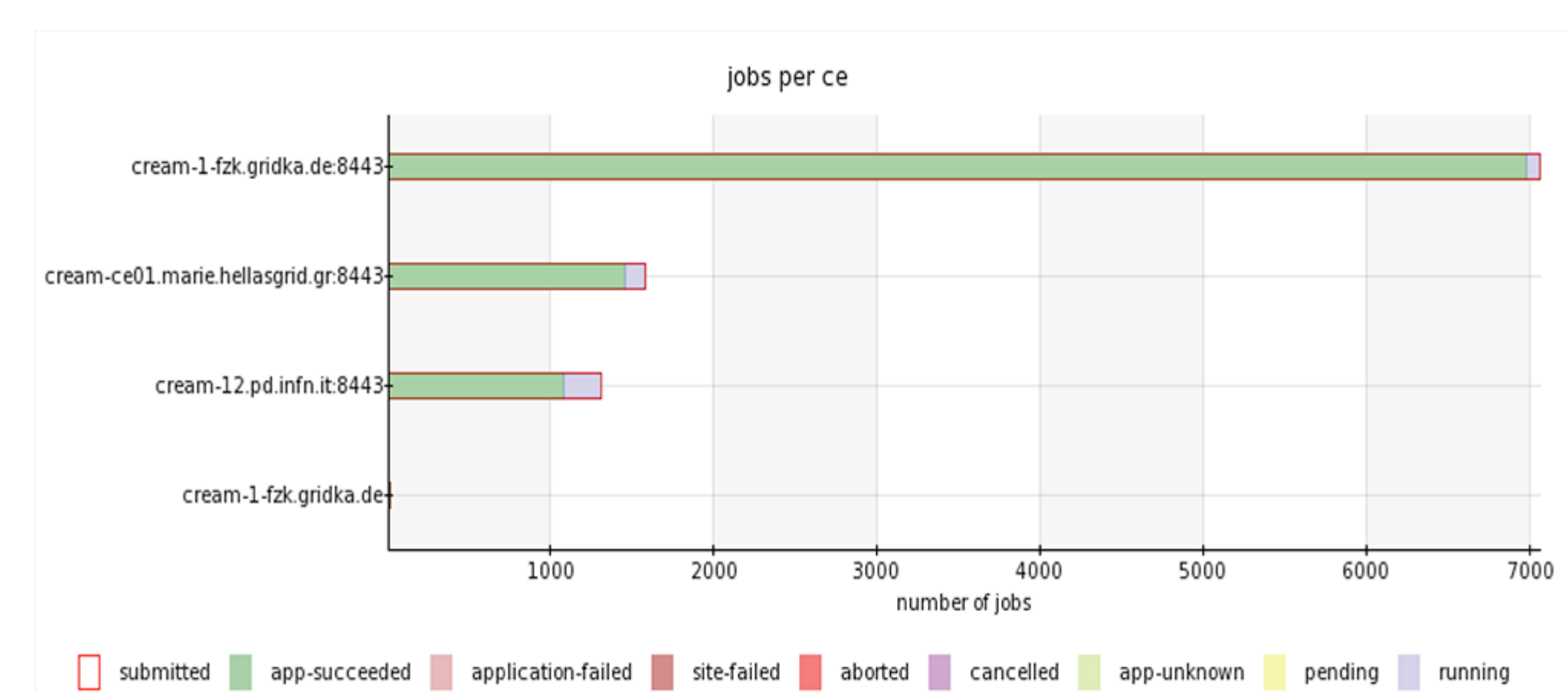
Glidein pilot jobs are sent to Grid sites using Condor-G



CMS tests with CREAM

glideinWMS Experience During CMS CCRC-08 sites across EGEE, OSG and Nordugrid
1st time Nordugrid ARC interface used in CMS

57 sites used between CCRC-08 and now.



Scalability tests

Condor scalability tested

Collector scalability

- on one machine, ~2GB RAM
- 25k in top-level collector
 - 70 sub-collectors handle authentication over WAN (between Europe and USA)

CCB scalability

- 25k glideins served from one machine
 - using the same 70 sub-collectors
 - 125,000 securely registered daemons

Schedd scalability

- 22k running jobs under a single schedd
 - 64-bit machine, 16GB RAM
 - current limit** due to network port usage
- 200k idle jobs on a single schedd

glideinWMS scalability tested

VO Frontend scalability

- 200k idle and 23k running user jobs
- 20 schedds

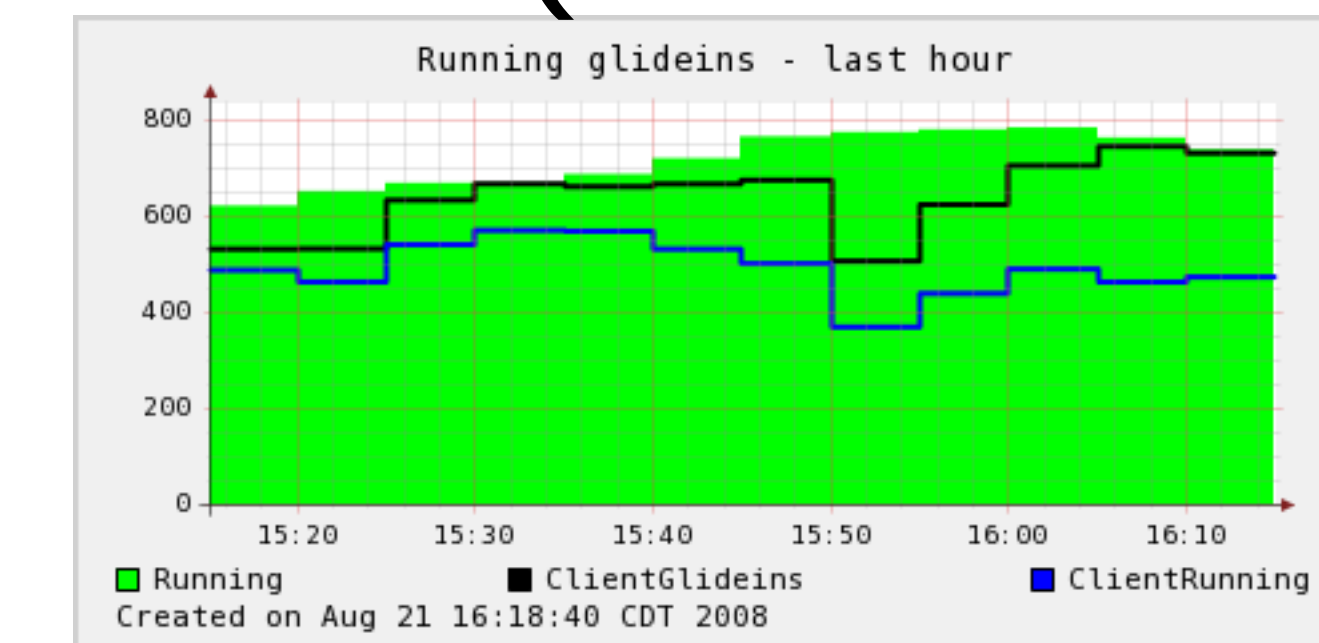
Glidein factory scalability

- 40 Grid sites **current limit**
- 100 Grid sites with some effort
- 5 VO frontends
- 23k running glideins

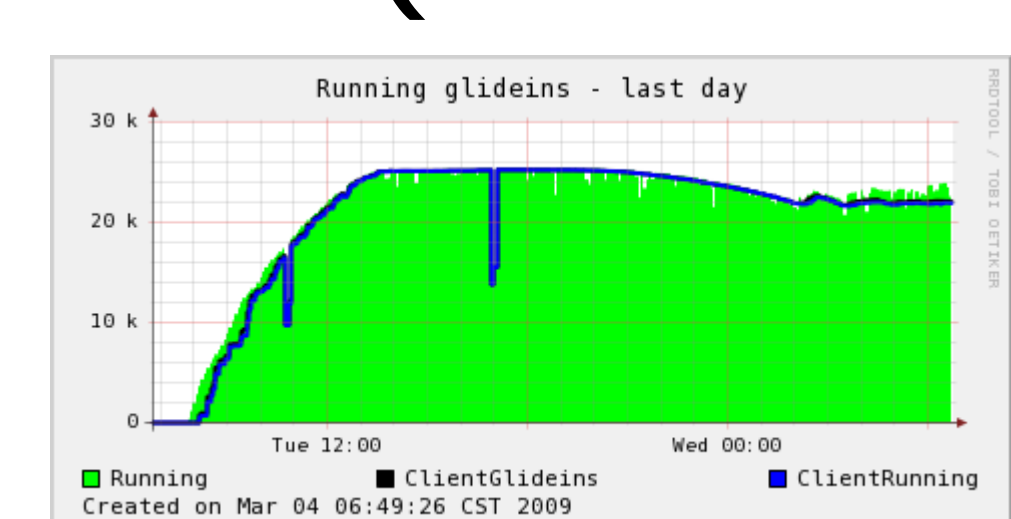
Prototype that scales higher available

GlideinWMS Cross-Atlantic Scale Tests:

Before: (Condor 7.1.2)



After: (Condor 7.3.1)



50 times better!