

eGEE

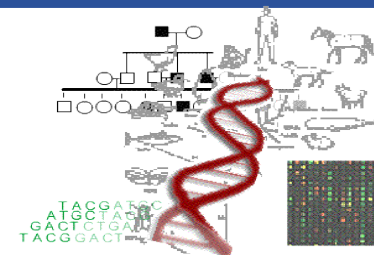
Enabling Grids for E-scienceE



Distributed Systems Laboratory



Technion



Computational Biology Laboratory

Superlink-online and BOINC

*Artyom Sharov, Mark Silberstein,
Dan Geiger, Assaf Schuster
CS Department, Technion*

www.eu-egee.org



Information Society
and Media



- **Purpose: to obtain crude chromosomal location of gene(s) associated with a phenotype of interest**
 - examples: Cystic fibrosis (found), diabetes, Alzheimer, blood pressure
- **We focus on parametric linkage analysis on pedigrees**
- **The analysis is very computationally intensive**
 - Inference in very large Bayesian networks

- **The system must be geneticists-friendly**
 - **On-line access** through simple WEB interface
 - **Interactive experience**
 - Low response time for short tasks
 - Fast computation of previously infeasible long tasks via parallel execution
 - Prompt user feedback
 - Secure, **reliable**, stable, overload-resistant
 - Allow submission of tasks by multiple users

- **User submits his/her data for analysis**

- No specification of running time or parallelization

- **Secured user web interface**

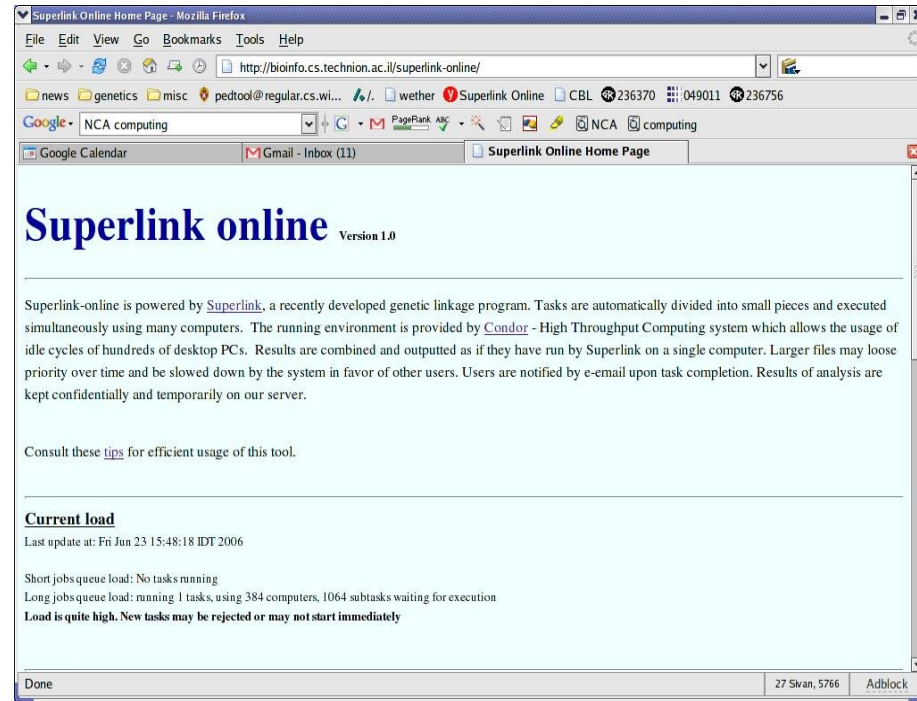
- Monitoring of partial results/running time expectations
- Cancellation

- **E-mail notifications on important events**

- Completion/Error/System failure

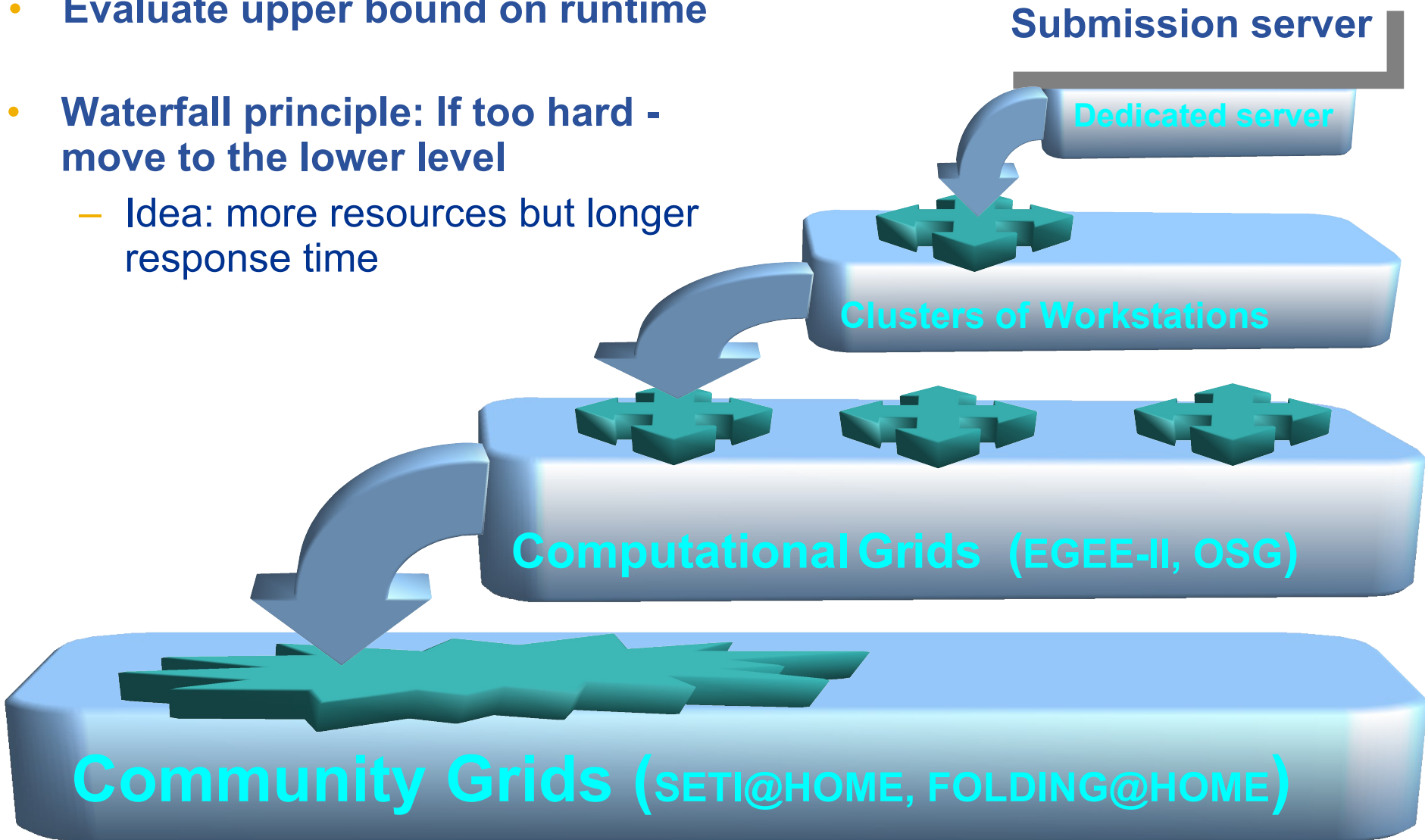
- ***Behind the scene***

- *Task runtime estimation and parallelization*
- *System monitoring, failure recovery*
- *Scheduling*



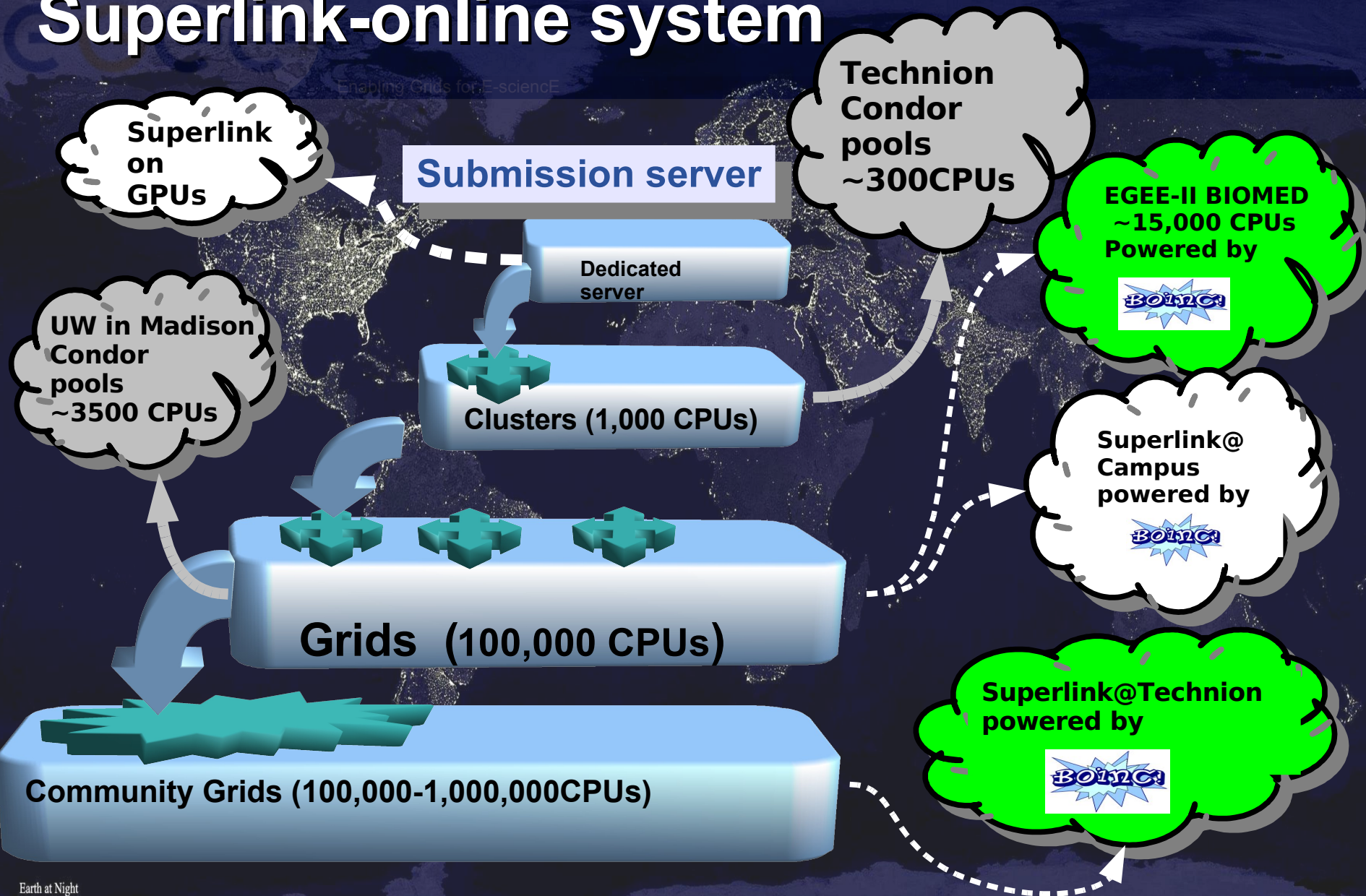
- ~240 **CPU years** for 12000 tasks during 2006-2007.
- **Several mutated genes found**
- **Users: Israeli and international users**
 - **NIH**, Universities and research centers in US, France, Germany, UK, Italy, Austria, Spain, Taiwan, Australia, and others...
- **Task example**
 - 250 days on single CPU -> 7 hours on ~300-700CPUS
 - Short tasks: few seconds even during severe overload

- Evaluate upper bound on runtime
- Waterfall principle: If too hard - move to the lower level
 - Idea: more resources but longer response time



Superlink-online system

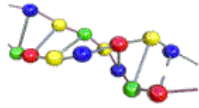
Enabling Grids for E-science



Earth at Night

<http://bioinfo.cs.technion.ac.il/superlink-online>

Welcome to **Superlink@Technion!**



Superlink@Technion helps geneticists all over the world find disease-provoking genes causing some types of diabetes, hypertension (high blood pressure), cancer, schizophrenia and many others. [Press here to learn more](#)

To contribute the idle cycles of your computer you need to [download](#) and install a small client, called BOINC. Please note that this project is still in **beta**, meaning that this site can be shut down for maintenance or work units may fail. We would appreciate your feedback via our [forums](#).

[Join now!](#)

Project status	
Account creation	enabled
Participants total	1238
Computers total	2263
Days of CPU utilized	4216.00
Detailed information	charts
Server state	state

System Requirements
<ul style="list-style-type: none"> RAM: 1GB of RAM is suggested, as the computation typically requires up to 700MB of memory. Disk space: insignificant. Supported platforms: Vista/XP/2000/NT and Linux. BOINC client version: 5.8.x is most preferable

User of the day



[Michael Casey](#)

I'm currently a senior in high school in California. I've been crunching for various projects for almost two years now, and I hope to one day find...

- **Parallel run – all results are required**
- **Workload: large memory footprint (400-700MB) WUs 3sec-1hour long, low I/O**
- **No checkpoints**
- **Multiple parallel tasks in FIFO**
 - Tasks of 100K to 10M WUs

- **Bumped into regular beginner's problems:**
 - credit hijacking, too short deadlines, too short WUs, server crash...
- **One month of operation:**
 - 767 active users, 1380 active hosts
 - ~10 CPU years on production tasks
- **Contributors are great!!!**



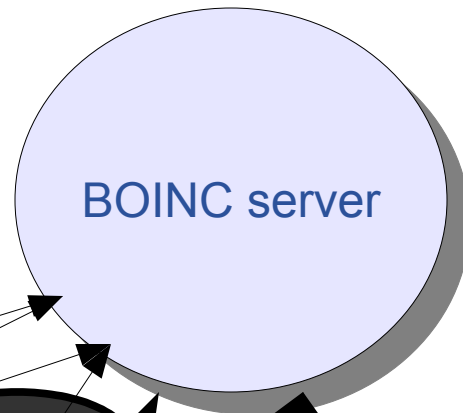
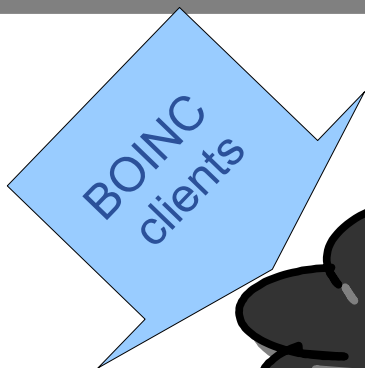
Enabling Grids for E-scienceE

BOINC as a generic engine for GRID submission

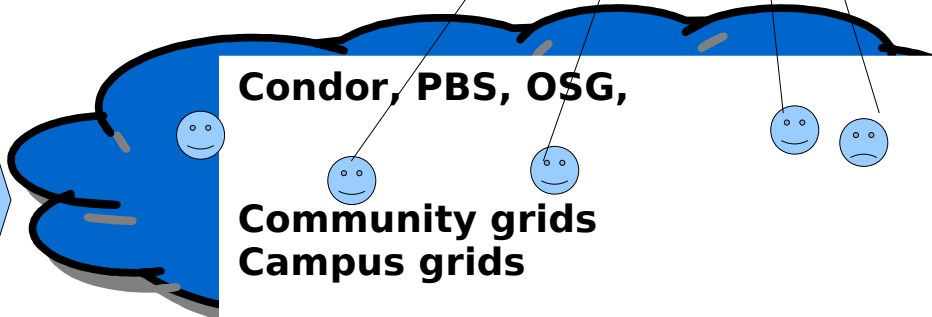
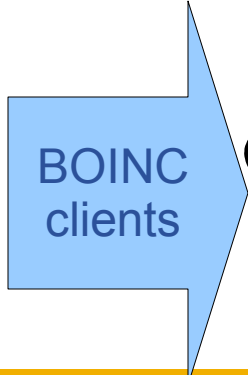
- **Same idea under different names: Master-worker/ application level scheduling/pilots/glide-ins**
 - Send BOINC clients as batch jobs, which then execute our jobs
 - **Decouples resource allocation from internal application scheduling**
 - Hides scheduling latency, failures, complexity
 - Builds your own on-demand cluster
 - **Decouples batch system specifics from application**
- **Why BOINC**
 - Fault tolerance, reliability, security, scalability, advanced scheduling mechanisms, “push” firewall-friendly, efficiency, built for opportunistic environments
 - Unifies community grids/standard grids/clusters at no development cost
 - Easy statistics/logging available

BATCH System specific submission machine

- Maintains required number of clients
- Restarts/reschedules/handles failures



BATCH System specific submission machine 2



Superlink@EGEE queue

Project status	
Account creation	disabled
Participants total	1
Computers total	16391
Days of CPU utilized	4317.84
Detailed information	charts
Server state	state

Community

- ◆ [Participant profiles](#)
- ◆ [Message boards](#)
- ◆ [Questions and answers](#)

Project totals and leader boards

- ◆ [Top participants](#)
- ◆ [Top computers](#)
- ◆ [Top teams](#)
- ◆ [Other statistics](#)



News

August 19, 2007 **Restoring from backup**

Everything restored back to the state as it was before the crash; plan for restoration from backup is prepared and rehearsed.

February 25, 2007 **Superlink compilation, job_status addons, started job_rm**

Statically compiled Superlink; finished PHP script for job_status; started CLI script for job_rm.

February 23, 2007 **job_status addons**

Finished CLI script for job_status.

February 21, 2007 **one_result_per_user_per_wu, GridWay scheduler options, job_status addons**

Eliminated the one_result_per_user_per_wu configuration option, which caused BOINC jobs not to be stuck, long runs take about 2-2.5 times less than before; playing with GridWay scheduler CLI parameters to prevent jobs from falling to bad GridWay clusters; added a few more options to both CLI and PHP interfaces of job_status script.

February 18, 2007 **Second long run, automatic scripts for cleaning and feeding GridWay queue, CLI for job_status**

Launched a second long run of 5000 workunits, 10 minutes each, in EGEE; in order to keep the queue busy all the time, wrote two scripts for cleaning the queue from finished jobs and feeding more jobs, when the number of queue jobs drops beyond 200; started preparing CLI for job_status (in Perl). [...more](#)

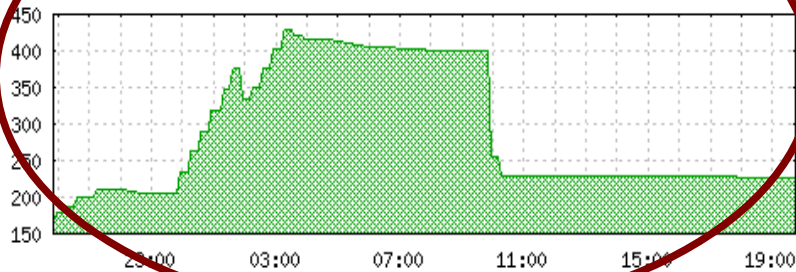
News is available as an [RSS feed](#) [XML](#).

- **EGEE results are VERY promising**
 - 10000 CPU hours (444 CPU days) in about 6 days
 - 100,000 CPU hours in about 2 weeks!
 - 55,000 jobs from 3sec-30min in 2 days on ~250 machines
 - Insight
 - EGEE environment is chaotic in the beginning, but at some point reaches steady state: BOINC clients find the machines which run them for days
 - Avoid Resource Broker overload -submit clients gradually
 - Automatic client termination if no WUs available
 - BOINC restricted to 1 core to comply with batch system allocation
 - Static compilation

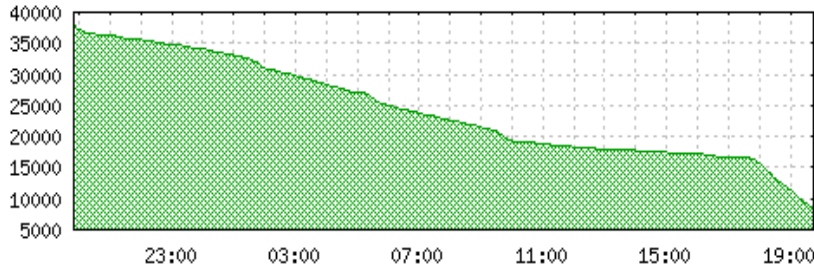
Server status

EGEE jobs status

Monitored jobs in submission machine (08/30/07 19:48 -> 08/31/07 19:48)
 (cur: 227.00 max: 428.00 min: 167.00 avg: 287.42)



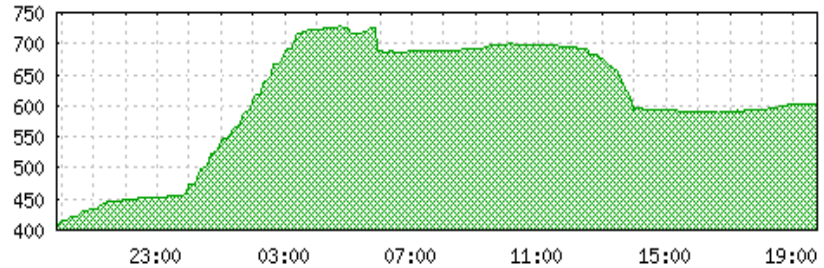
Number of unfinished workunits (08/30/07 19:48 -> 08/31/07 19:48)
 (cur: 8292.00 max: 38056.00 min: 8292.00 avg: 24279.94)



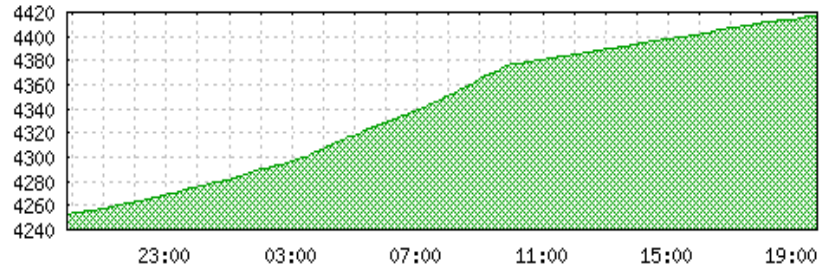
WU processing rate

Last BOINC clients status

Number of active hosts (08/30/07 19:48 -> 08/31/07 19:48)
 (cur: 603.00 max: 728.00 min: 405.00 avg: 610.86)



Days of CPU utilized (08/30/07 19:48 -> 08/31/07 19:48)
 (cur: 4417.44 max: 4417.44 min: 4252.95 avg: 4341.68)



- **Allow easy monitoring and submission of BOINC WUs from scripts**
 - boinc_status
 - job_rm
 - job_status
- **Assimilator emulates regular Batch system**
 - Returns the files back to the submission directory
- **Currently application specific, but can be easily extended**

- **Problem with BOINC on Campus grids**
 - BOINC does not monitor the resource during the run
 - Causes contention with periodic tasks (Windows/Antivirus updates) and local nightly runs
- **LoadD Condor-based daemon**
 - Solves the problem of load monitoring during the execution
 - Required for successful campus deployment of BOINC
 - See

<http://dsl.cs.technion.ac.il/projects/gozal/developers/loadd.zip>

- **Host groups**
 - Prefer predefined hosts over others
- **Multicore scheduling model on the client is problematic for high memory footprint WUs**
 - Cores stay idle because of memory
- **Result validation without duplication of WUs**
 - We know the problem is tough... but working on it

- **Anna Tzemach, Julia Stolin, Edward Vitkin, Nikolay Dovgolevsky, Maayan Fishelson**
- **Generous support of Prof. Livny by allowing us to use Condor pool resources in UW Madison**
- **Condor team support**

Visit us at:

**[http://bioinfo.cs.technion.ac.il/
superlink-online](http://bioinfo.cs.technion.ac.il/superlink-online)**