"An Evaluation Methodology for Computational Grids"

Eduardo Huedo (huedoce@inta.es)

Rubén S. Montero

Ignacio M. Llorente

Advanced Computing Laboratory

Centre for Astrobiology (INTA-CSIC)

Distributed Systems Architecture and Security group

Complutense University of Madrid

HPCC 2005 Conference

Objectives

- Propose an appropriate set of **criteria** and **metrics** which allow evaluating the capabilities of a computational Grid environment from a **user's point of view**.
- Apply these criteria and metrics in the evaluation of a Grid environment, based on:
  - **Globus** basic services, and
  - **Grid*Way*** submission framework,

  using **NGB**, implemented in **DRMAA**, as test programs.
- Note that **NGB** and **DRMAA** are **not tied** to any specific Grid middleware or submission framework.

Evaluation Criteria

- Initially, we propose **functionality**, **reliability** and **performance** as general criteria to evaluate a Grid environment from a user's point of view.
  - Why these criteria? The focus is on performance. However, in the current state of Grid technology, functionality is usually limited, and reliability is often the weak link in system performance.
  - Nevertheless, in the future, other criteria could be addressed.
- We have tried to keep the evaluation criteria **simple** and **objective**. In this sense, each metric should be:
  - **easy to measure** (directly provided by typical submission frameworks), and
  - **easy to compare** (having a Boolean or numeric value).

Functionality Criterion

- Ability to execute unattended distributed communicating applications (i.e. the **NGB** suite).
  - This is an implicit requirement, but it is worth to mention it given the current status of Grid computing technologies.
- Support for standard high-level interfaces, like **DRMAA**.

Reliability Criterion

- A job should, transparently to the user, continue its execution (at least from the beginning) in other resource when some of the following failure or loss of quality of service conditions take place:
  - **job cancellation** (failure) or **suspension** (QoS loss),
  - **system crash** (failure) or **saturation** (QoS loss), and
  - **network disconnection** (failure) or **saturation** (QoS loss).

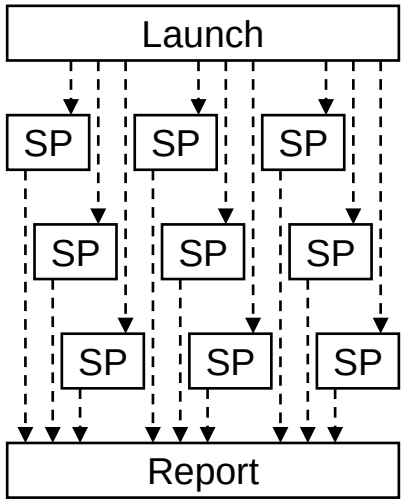
Performance Criterion

- **User-level** metrics:
  - **Turnaround time** (**T**),
  - **Productivity** (**P**).
- **Diagnostic** and **tuning** metrics:
  - **Response time** ($\mathbf{T_r}$),
  - **Transfer time** ($\mathbf{T_{xfr}}$),
  - **Execution time** ($\mathbf{T_{exe}}$),
  - **Resource usage** (**U**).

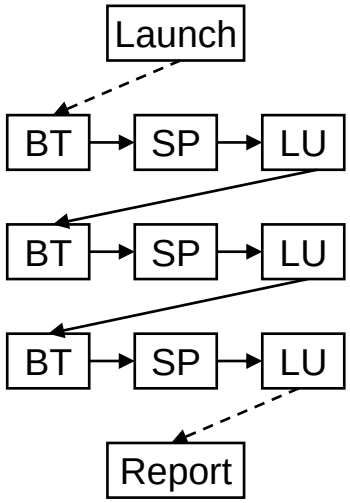
Grid Benchmarks

- **Data flow graphs** encapsulating an instance of a **NPB** code in each graph node, which communicates with other nodes by sending or receiving initialization data.
- The **NPB** codes symbolize **scientific computation** (flow solvers SP, BT and LU), **post-processing** (data smother MG) and **visualization** (spectral analyzer FT).

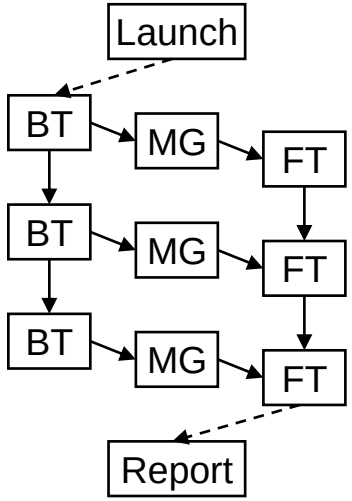
Embarrassingly Distributed (ED)



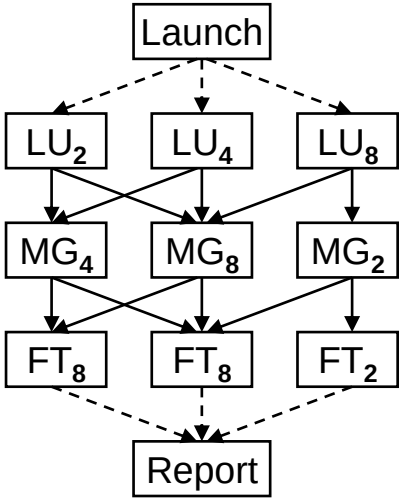
Helical Chain (HC)



Visualization Pipe (VP)



Mixed Bag (MB)



Submission Framework

Easier and efficient execution in dynamic and heterogeneous grids in a ***submit & forget*** fashion.

Functionality:

- **Adaptive scheduling**
- **Adaptive execution**
- **Fault tolerance**

www.gridway.org

Design guidelines:

- **Adaptable/extensible** (modular design)
- **Scalable** (decentralized architecture)
- **Deployable** (user, standard services)
- **Applicable** (wide application range)

Research Testbed

- Based on **Globus** pre-WS services.
- **Globus** WS services are also supported now.
  - Now, we have the opportunity to apply again the methodology and compare results.
- The small size of the testbed is not an issue (at least not a big one).

Name	Processor	Speed	OS	Mem.	DRMS
pegasus	Intel Pentium 4	2.4GHz	Linux 2.4	1GB	fork
hydrus	Intel Pentium 4	2.5GHz	Linux 2.4	512MB	fork
cygnus	Intel Pentium 4	2.5GHz	Linux 2.4	512MB	fork
cepheus	Intel Pentium III	600MHz	Linux 2.4	256MB	fork

Functionality Results

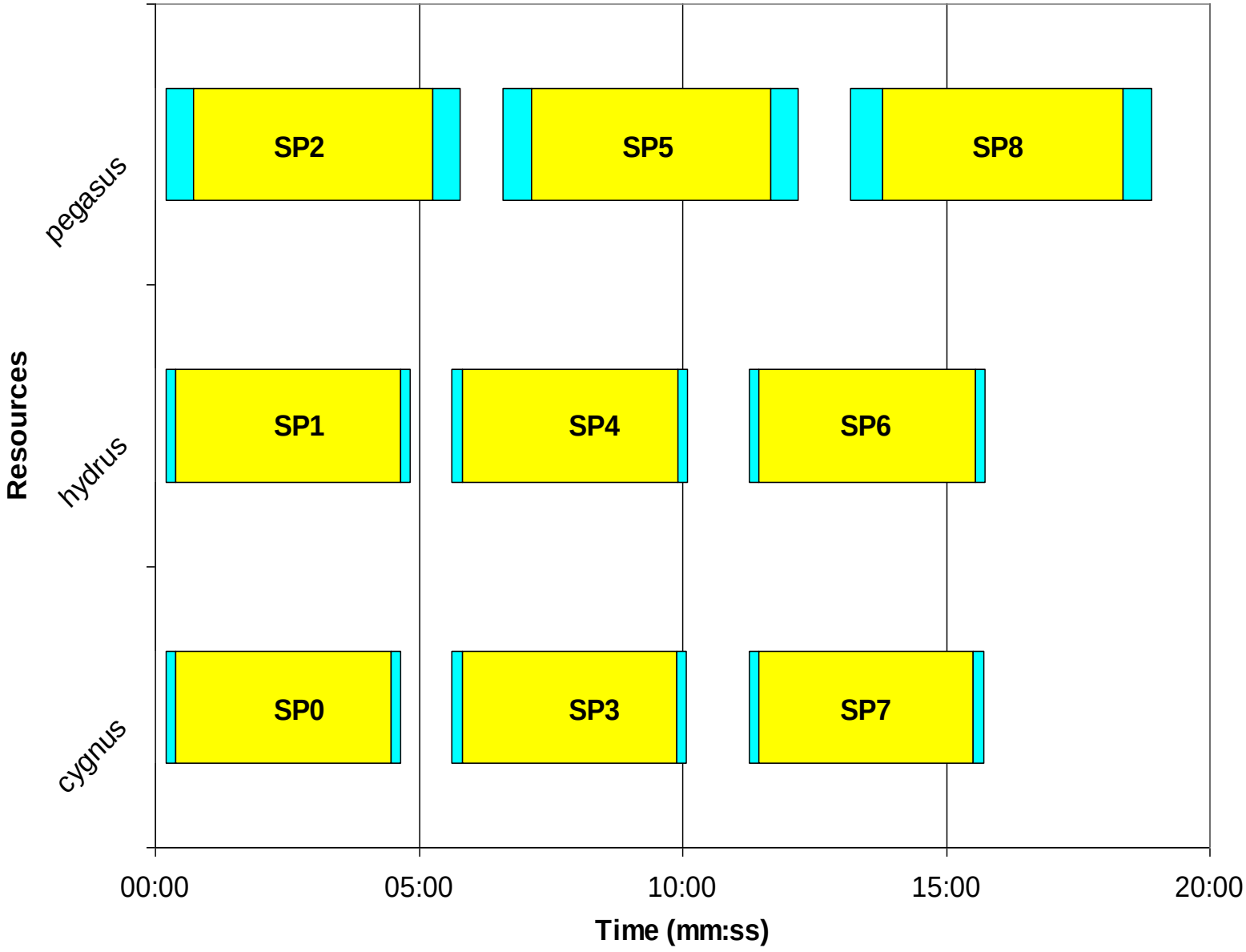
- The paper-and-pencil specification of the **NGB** suite has been fully implemented by using the **DRMAA** interface supported in **Grid*Way***.

```
/* Initialization */
jt = SP;
num_jobs = 9;
drmaa_init(contact, err);

/* Submit all jobs simultaneously and wait for them */
drmaa_run_bulk_jobs(&job_ids, jt, 0, num_jobs-1, 1,
    err, DRMAA_ERROR_STRING_BUFFER);
drmaa_synchronize(job_ids, DRMAA_TIMEOUT_WAIT_FOREVER,
    1, err, DRMAA_ERROR_STRING_BUFFER);

drmaa_exit(err);
```

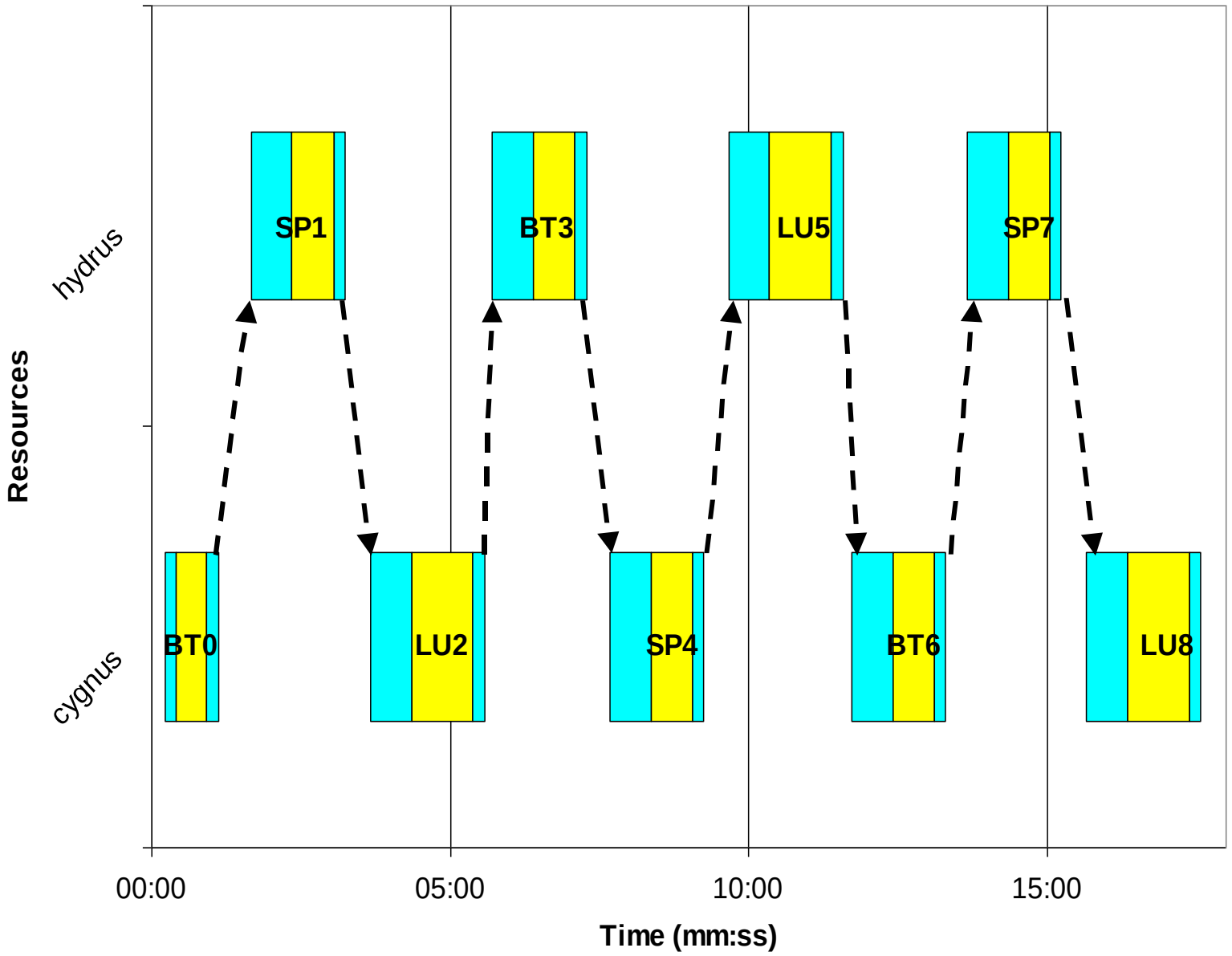
Reliability Results

- The **HC** benchmark constitutes an excellent probe to evaluate the reliability criterion, since output files of each task can be used as **checkpoints** for the next task.
- Failure or QoS loss conditions **artificially generated**:
  - **Grid*Way*** detects these conditions and triggers a **job migration**, however
  - it does not consider **network saturation** as a failure condition, instead, it uses network status information to rank resources.

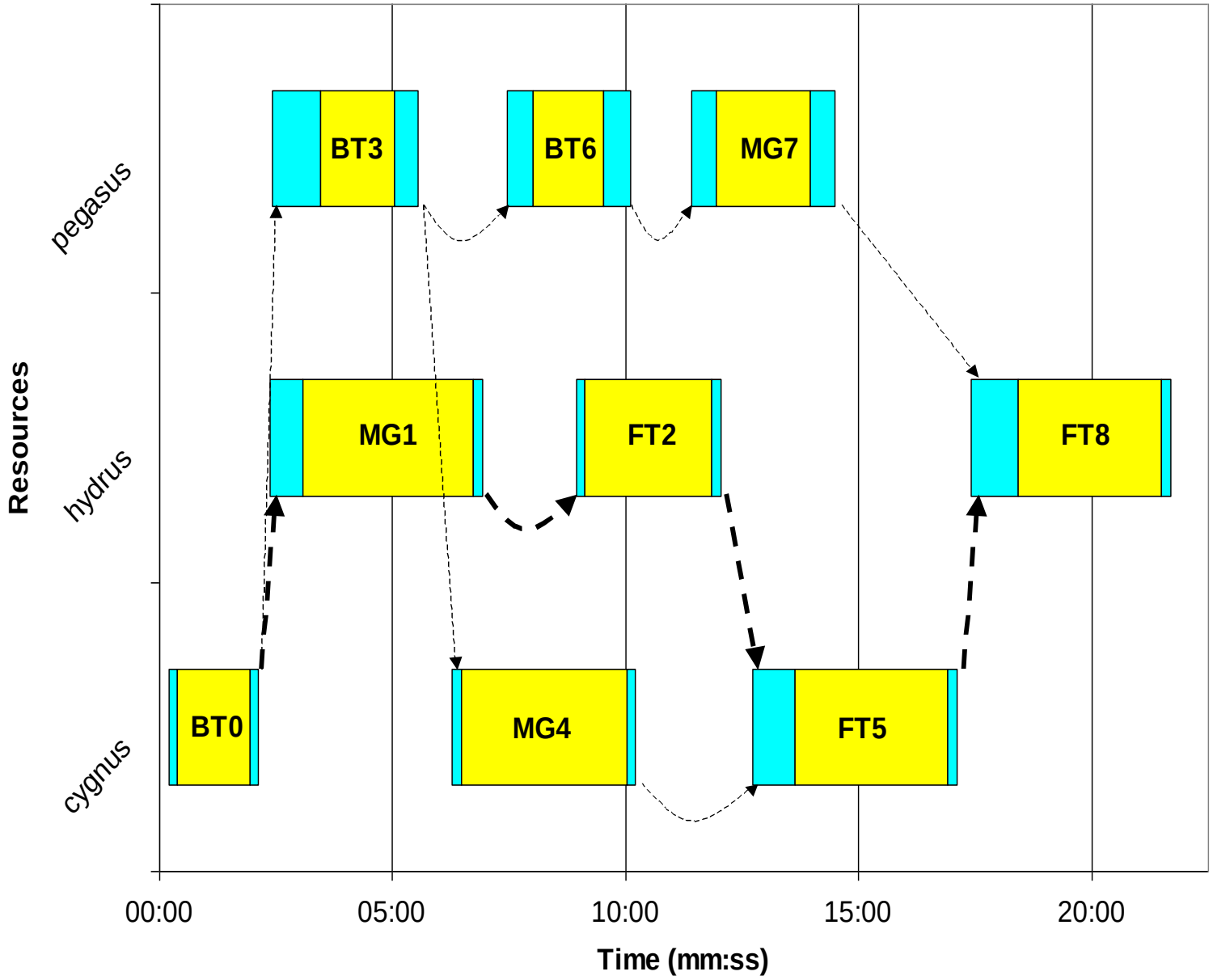
Performance Results: ED.A Execution Profile



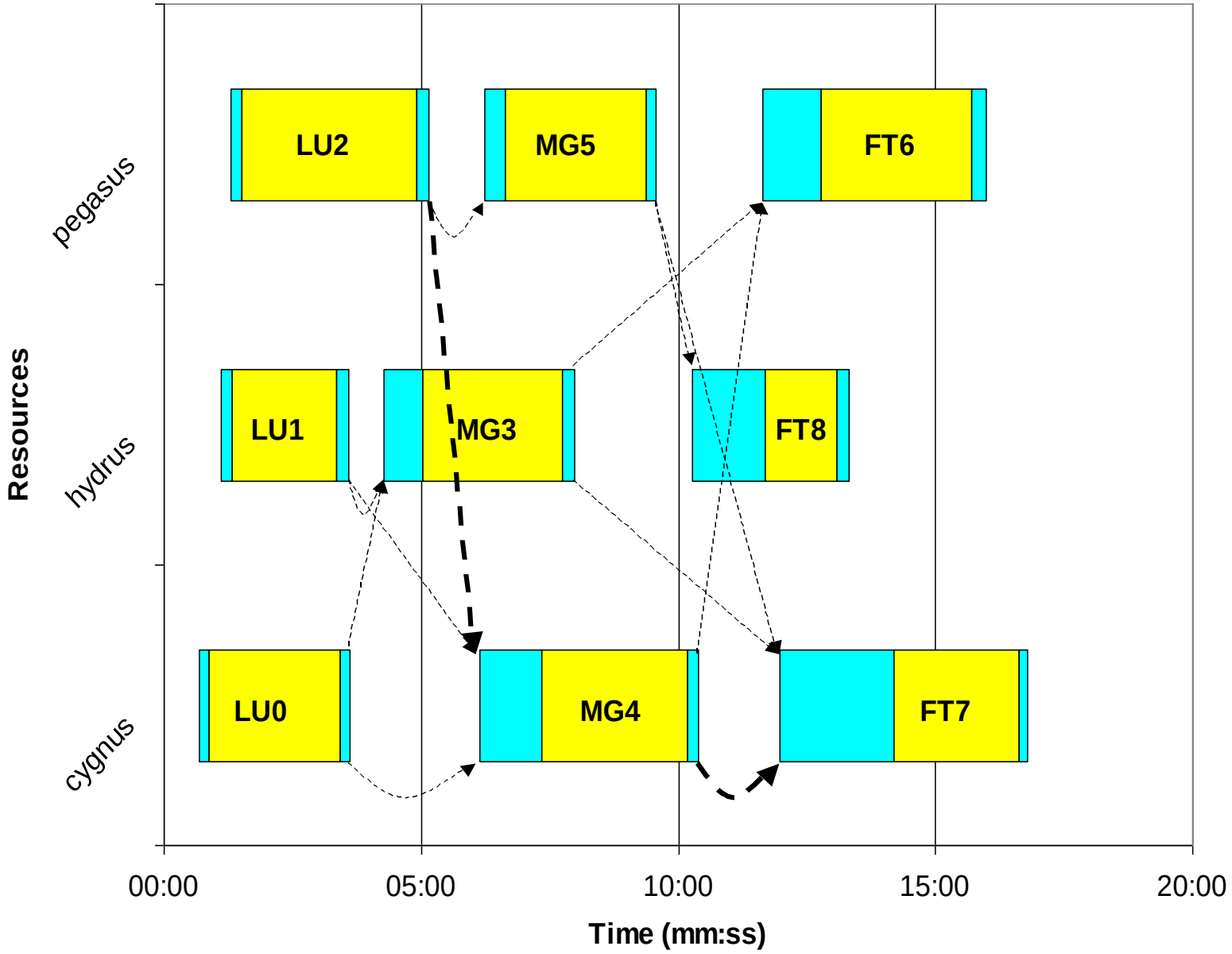
Performance Results: HC.A Execution Profile



Performance Results: VP.A Execution Profile



Performance Results: MB.A Execution Profile



Performance Results: Summary

Metric	Description	Unit	Benchmark			
			ED.A	HC.A	VP.A	MB.A
T	Turnaround time	minutes	18.88	17.57	21.67	16.80
P	Productivity	jobs/hour	28.60	30.73	24.92	32.14
T_r	Response time	minutes	-	3.09	-	-
T_{xfr}	Transfer time	minutes	5.50	7.10	8.10	9.70
T_{exe}	Execution time	minutes	38.30	7.38	22.93	23.03
U	Resource usage	-	2.02	0.42	1.06	1.37

Conclusions

- The proposed (incomplete) evaluation methodology could be useful to:
  - **validate** the infrastructure, middleware and submission framework,
  - **adjust** their components,
  - **compare** alternative implementations, and
  - **estimate** the achieved quality of service.
- Each benchmark family has different characteristics:
  - **ED** is good to measure **raw performance** (throughput) and study **variability** (dynamic behaviour) in transfer and execution times,
  - **HC** is good to measure **reliability** and **overheads**, and
  - **VP** and **MB** are good to evaluate **functionality** and different **scheduling strategies**.
- We found several ways to improve **Grid*Way***, mainly reducing the time between tasks sequentially submitted to the same resource (T_r).
- The research testbed is rather small, but more resources would only benefit benchmark **ED**. **Scalability** is limited by the application.
- We are now testing on a **bigger testbed** with a **modified NGB suite**.