Simulation of Mars Impact Cratering on a Grid Environment

Eduardo Huedo (huedoce@inta.es)
Alain Lepinette
Rubén S. Montero
Ignacio M. Llorente
Luis Vázquez

Advanced Computing Laboratory
Centro de Astrobiología
Associated to *NASA Astrobiology Institute*
CSIC – INTA

Distributed Systems Architecture & Security Group
Dpto. de Arquitectura de Computadores
Universidad Complutense de Madrid

GADA 2004

Motivation and Goal

- Introduce the simulation of **Mars impact cratering**.
- Present another case of an application successfully executed on the Grid (i.e. just another *"Making something on the Grid"* paper).
- Demonstrate the suitability, in terms of performance and fault tolerance, of **The Grid*W*ay Framework** to run large-scale computational experiments.
- Bring up the benefits of using a **Grid environment**.

Outline

1. Simulation of Mars Impact Cratering
2. The Grid*W*ay Framework
3. Experimental Testbed
4. Results
5. Conclusions

Simulation of Mars Impact Cratering

- Impact cratering is a geological process of special interest in Astrobiology that affects the surface of nearly all celestial bodies.
- Marine-target impact cratering simulation plays an important role in the study of past martian seas. A water layer at the target influences lithology and morphology of the resultant crater.
- Astrobiologists want to analyze the threshold impactor diameter for cratering the seafloor of an hypothetical martian sea.
- The search space of input parameters includes the projectile diameter itself (8 cases), the water depth (3 cases) and the impactor velocity (3 cases).

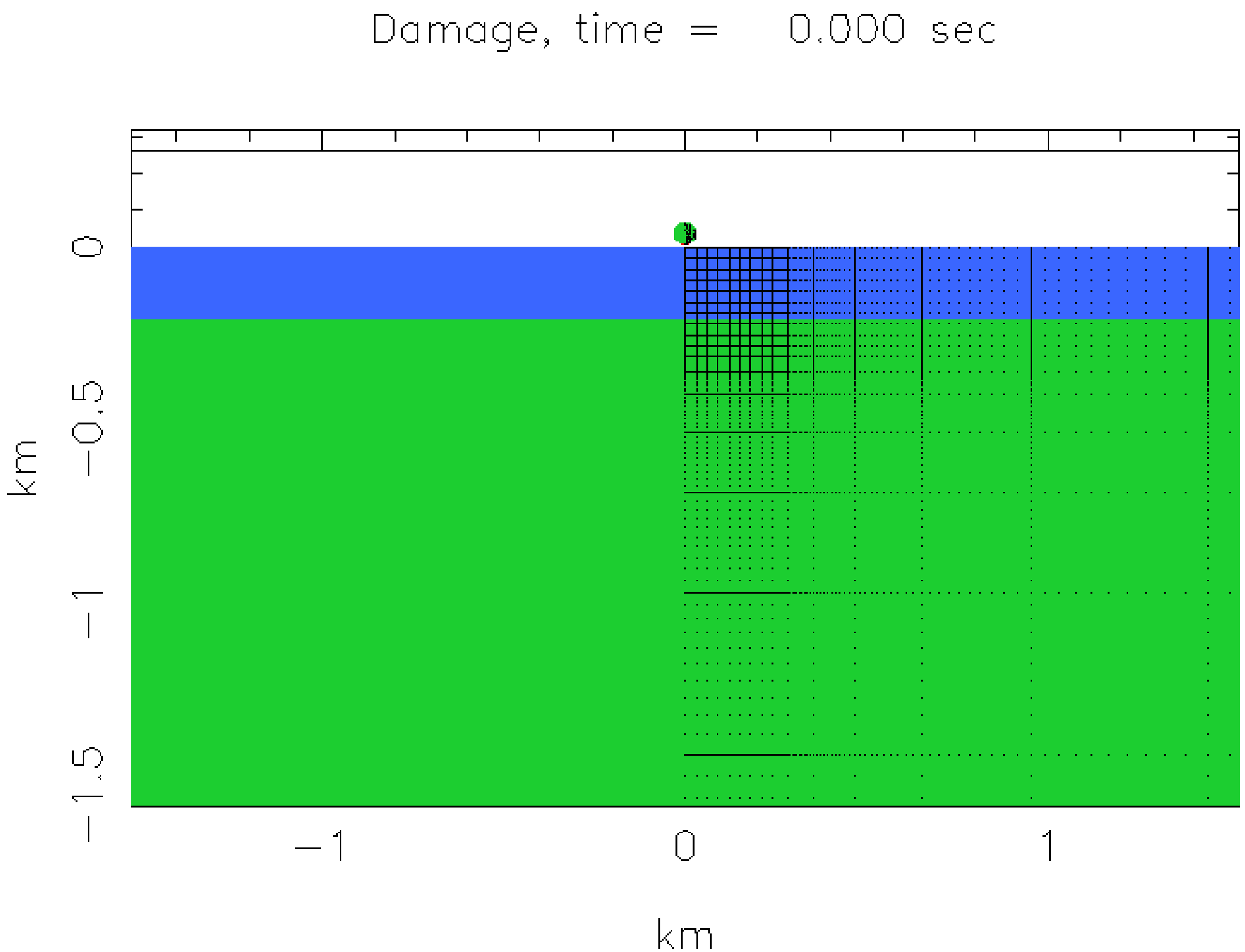
Simulation of Mars Impact Cratering

- The impact process can be described as a transfer of energy process: the initial kinetic energy of the projectile creates a hole (the crater) and heats the material of both projectile and target.

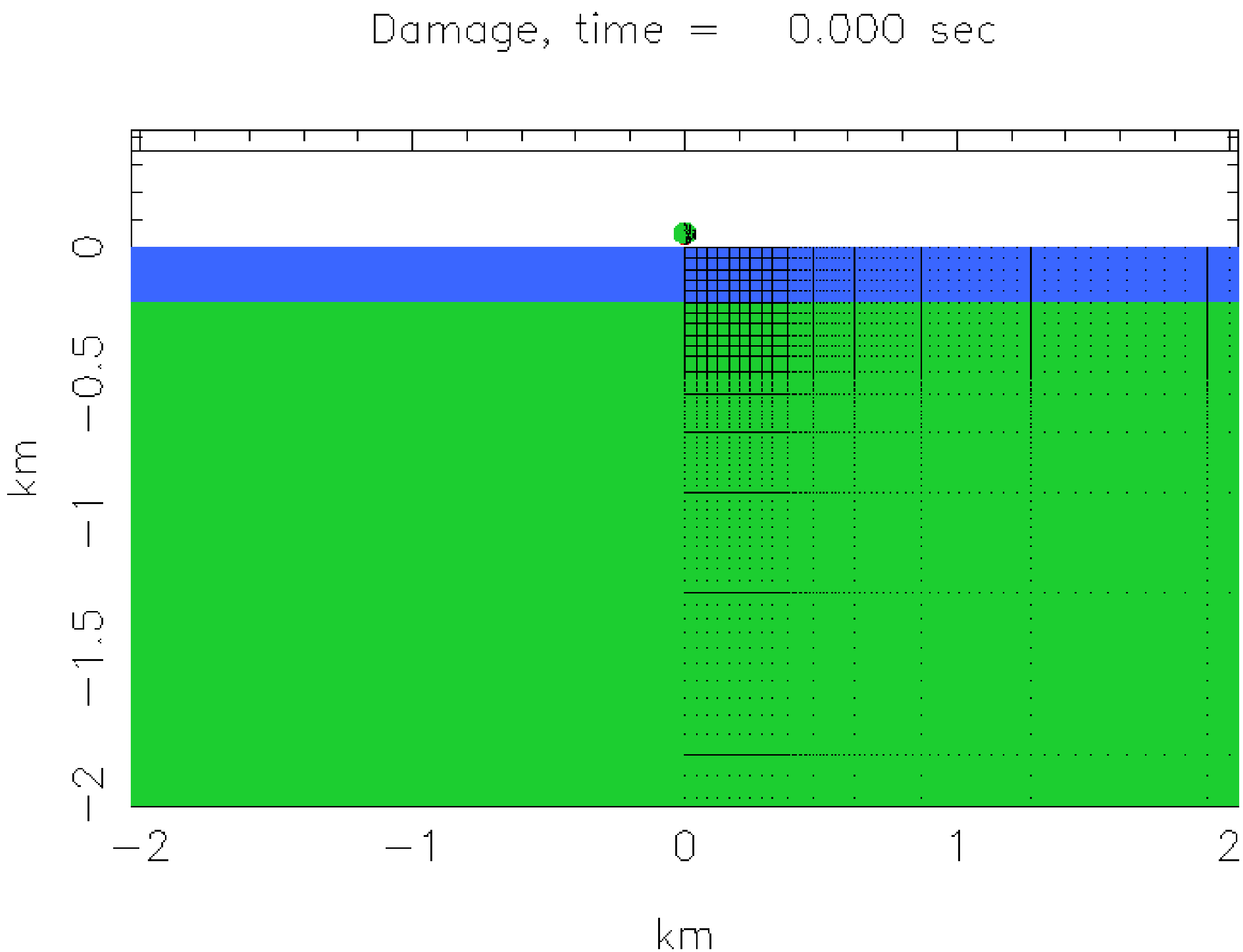
- High-velocity impacts have the following main stages:
  1. contact and shock compression
  2. transient cavity growth
  3. crater material ejection
  4. transient cavity modification

- 2D hydrocode based on SALE (Simplified Arbitrary Lagrangian-Eulerian) to solve the equations of motion (Navier-Stokes for compressible media) and estate.

D= 60m, H= 200m, V= 10Km/s



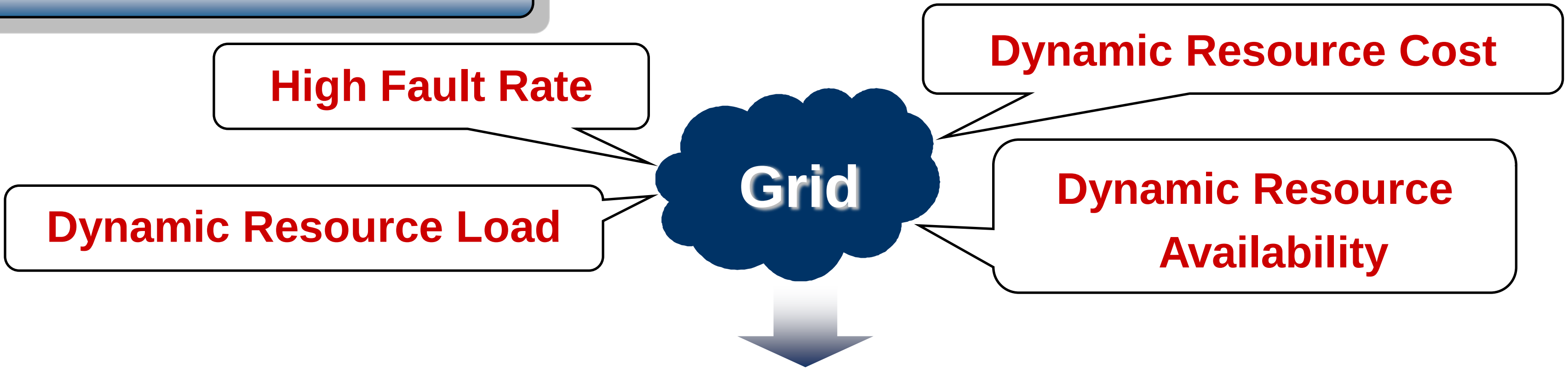
D= 80m, H= 200m, V= 10Km/s



User on a Globus Grid

Where do I execute my job?	**resource selection**
What do I need (files...)?	**job preparation**
How do I execute my job?	**job submission**
How is my job doing?	**job monitoring**
Can I use a better host?	**job migration**
How do I retrieve job output?	**job termination**

Grid Characteristics



Job must be able to **migrate** among grid resources to obtain **application performance** and **fault tolerance**

Philosophy

Provides an **easier** and more **efficient** execution of jobs (***submit*** *&* ***forget***) on ***heterogeneous***, ***dynamic*** and ***loosely coupled*** Grids.

Design Guidelines

- **Easy to Apply** (legacy applications)
- **Easy to Adapt** (modular design)
- **Easy to Scale** (decentralized architecture)
- **Easy to Deploy** (user, standard services)

Features

- Adaptive scheduling
- Adaptive execution
- Self-adaptive applications
- Fault-tolerance

User Interface (UNIX-like)

- **gwsubmit:** submits a job, or an array job (like a fork)
- **gwps:** displays job information and status

JID	AID	TID	DM	SM	GSM	STIME	ETIME	EXETIME	XFRTIME	EX	TEMPLATE	HOST
0	--	--	zomb	done	--	13:51:41	14:07:29	0:15:25	0:00:23	--	crat.job	babieca/pbs
1	--	--	subm	wrap	actv	14:03:41	--:--:--	0:03:40	0:00:58	--	crat.job	cygnus/fork
2	--	--	subm	prol	actv	14:07:57	--:--:--	0:00:00	0:00:22	--	crat.job	hydrus/fork

- **gwhistory:** displays job execution history

REASON	STIME	ETIME	EXETIME	XFRTIME	RANK	HOST
disc	14:03:41	14:05:29	0:01:25	0:00:23	-630	cepheus.dacya.ucm.es/jobmanager-fork
--	14:05:29	--:--:--	0:02:15	0:00:35	-420	cygnus.dacya.ucm.es/jobmanager-fork

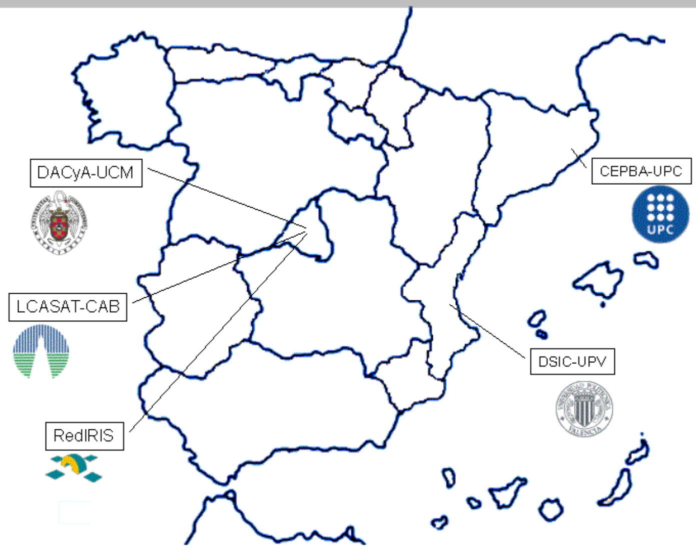
- **gwkill:** signals a job (kill, stop, resume, reschedule)
- **gwwait:** waits for zombie state of a job (any, all, set)

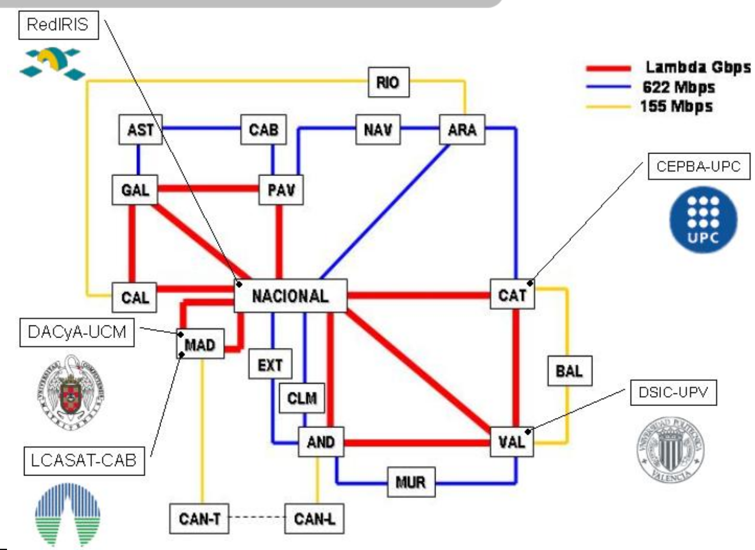
Client API (DRMAA subset)

Handles **job submission**, **monitoring** and **control**, and retrieval of **finished job status**.

Experimental Testbed

Geographical Distribution and Interconnecting Network

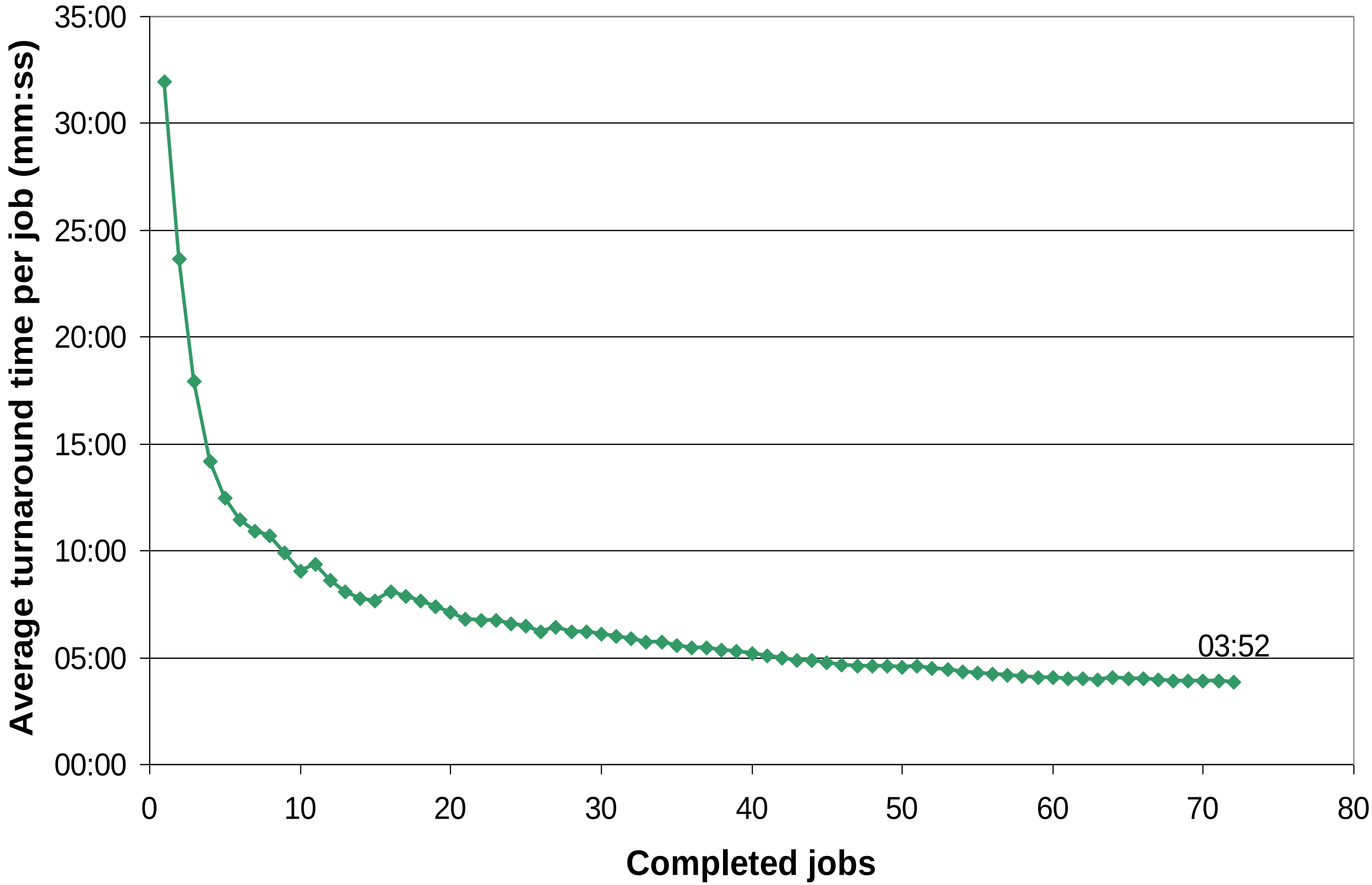




Resource Description

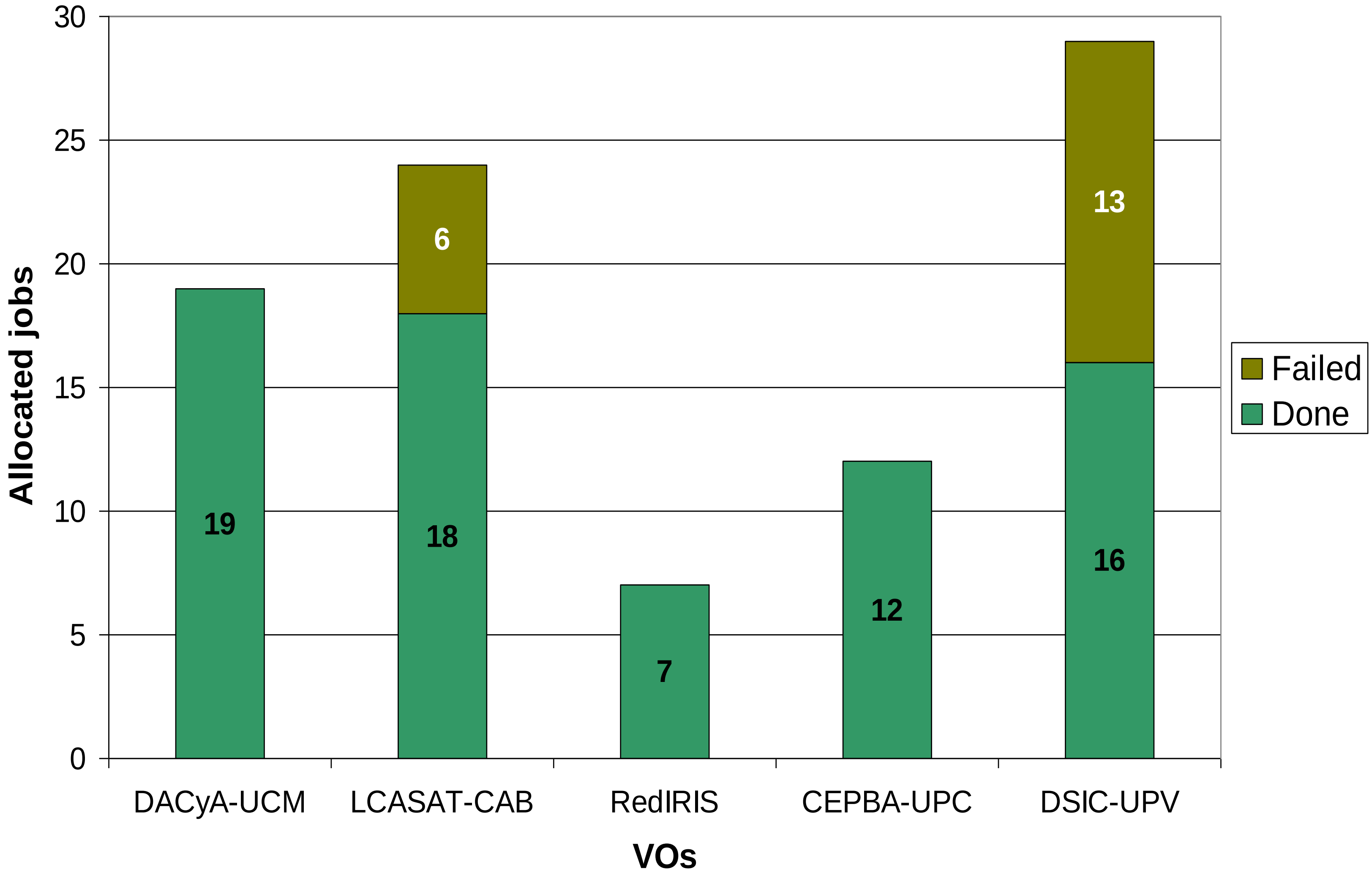
Name	Site	Nodes	Model	Speed	Mem	OS	Job mgr.
hydrus		1	Intel P4	2.5GHz	512MB	Linux 2.4	fork
cygnus	DACyA-UCM	1	Intel P4	2.5GHz	512MB	Linux 2.4	fork
aquila		1	Intel PIII	700MHz	128MB	Linux 2.4	fork
babieca	LCASAT-CAB	5	Alpha EV67	450MHz	256MB	Linux 2.2	PBS
platon	RedIRIS	2	Intel PIII	1.4GHz	512MB	Linux 2.4	fork
heraclito		1	Intel Celeron	700MHz	256MB	Linux 2.4	fork
ramses	DSIC-UPV	5	Intel PIII	900MHz	512MB	Linux 2.4	PBS
khafre	CEPBA-UPC	4	Intel PIII	700MHz	512MB	Linux 2.4	fork

Results



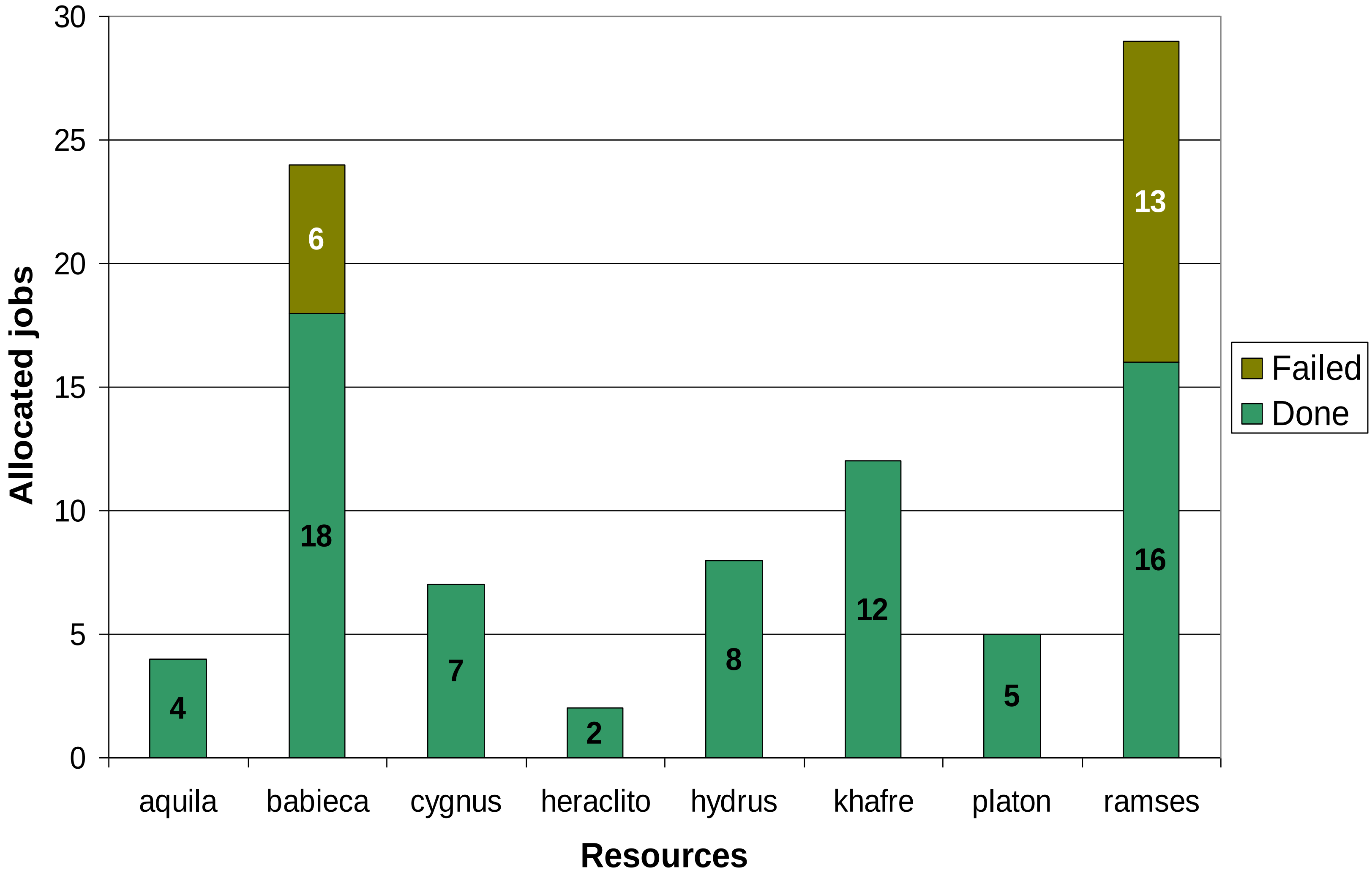
Average turnaround time per job (mm:ss)
35:00
30:00
25:00
20:00
15:00
10:00
05:00
00:00
0
10
20
30
40
50
60
70
80
Completed jobs
03:52

Results



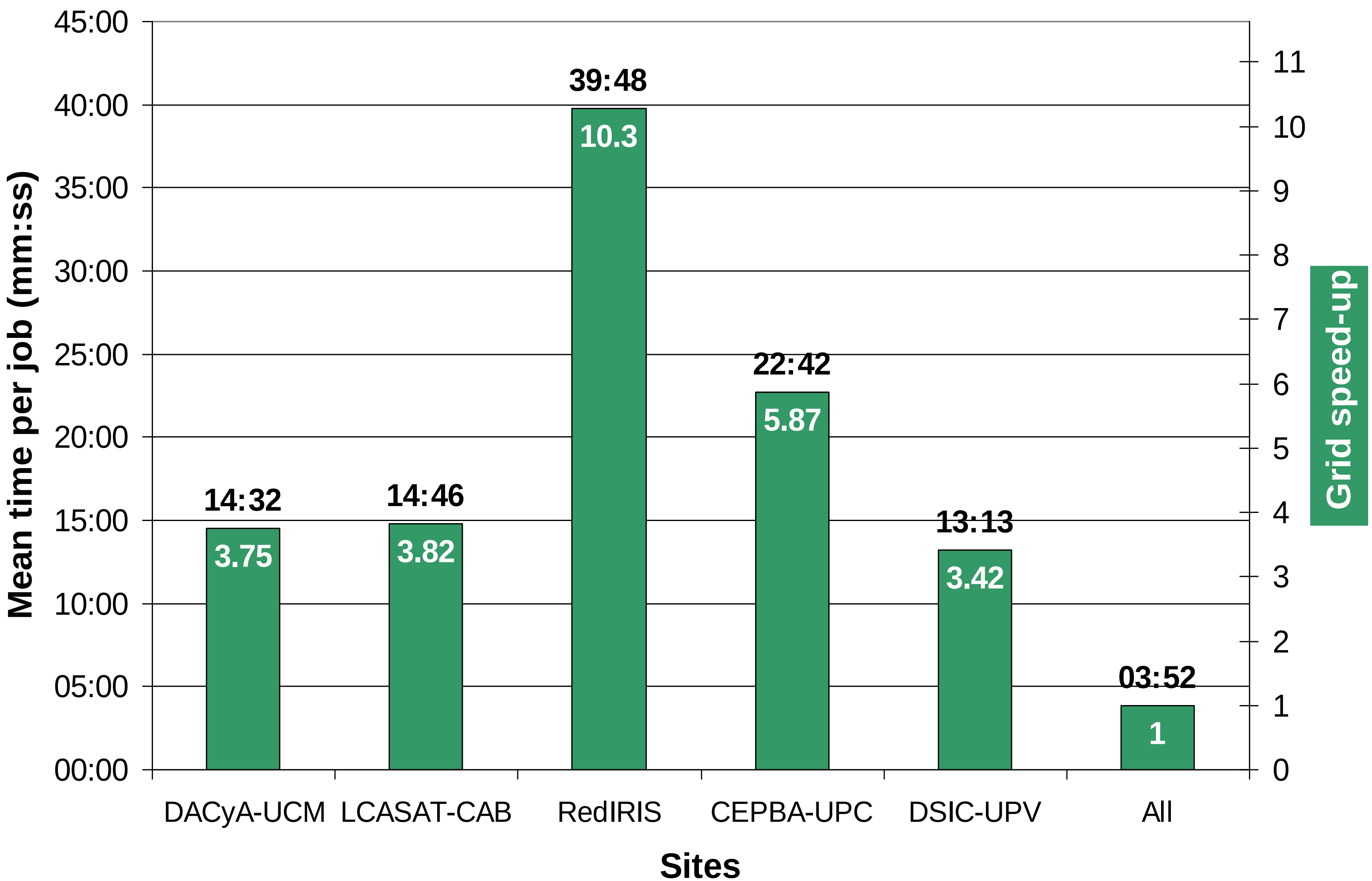
30
25
20
15
10
5
0
Allocated jobs
19
6
18
7
12
13
16
Failed
Done
DACyA-UCM
LCASAT-CAB
RedIRIS
CEPBA-UPC
DSIC-UPV
VOs

Results



30
25
20
15
10
5
0
Allocated jobs
6
13
18
16
4
7
2
8
12
5
aquila
babieca
cygnus
heraclito
hydrus
khafre
platon
ramses
Resources
Failed
Done

Results


45:00
40:00
35:00
30:00
25:00
20:00
15:00
10:00
05:00
00:00
Mean time per job (mm:ss)
11
10
9
8
7
6
5
4
3
2
1
0
Grid speed-up
39:48
10.3
22:42
5.87
14:32
3.75
14:46
3.82
13:13
3.42
03:52
1
DACyA-UCM
LCASAT-CAB
RedIRIS
CEPBA-UPC
DSIC-UPV
All
Sites

Conclusions

- Results of these analysis can be used:
  1. to develop a search criteria for future exploration missions (e.g. ground penetrating radar surveys included in ESA Mars Express and planned for NASA 2005 missions), and
  2. to understand the morphologies for future investigations.
- The discovery of marine-target craters on Mars would help:
  1. to address the ongoing debate of whether large water bodies occupied the northern plains of Mars, and
  2. to help constrain future paleoclimatic reconstructions.

Conclusions

- **Simulation studies** often require an amount of computing power that is not usually available at a single organization.
- **Grid technology** allows the federation of resources from different organizations, with respect for each site autonomy, to help in the construction of virtual organizations.
- However, **efficient** and **reliable** execution on Grids involves challenging issues.
- **The Grid*W*ay Framework** provides an easier and more efficient execution of jobs on dynamic and heterogeneous Grids.

Thank you
for your attention!

More information…

http://asds.dacya.ucm.es/GridWay

huedoce@inta.es