Simulation of Mars Impact Cratering on a Grid Enviroment*

Eduardo Huedo[1], Alain Lepinette[1], Rubén S. Montero[2], Ignacio M. Llorente[2,1], and Luis Vázquez[3,1]

[1] Laboratorio de Computación Avanzada, Simulación y Aplicaciones Telemáticas, Centro de Astrobiología (CSIC-INTA), Associated to NASA Astrobiology Institute, 28850 Torrejón de Ardoz, Spain.
[2] Departamento de Arquitectura de Computadores y Automática, Facultad de Informática, Universidad Complutense de Madrid, 28040 Madrid, Spain.
[3] Departamento de Matemática Aplicada, Facultad de Informática, Universidad Complutense de Madrid, 28040 Madrid, Spain.

Abstract. Marine-target impact cratering simulation plays an important role in the study of past martian seas. In order to develop profiles for future exploration missions and to understand the morphologies for future investigations, a large number of simulations have to be done. This paper presents some experimental results obtained with the execution of impact cratering simulations in a Globus-based Grid environment through the Grid*W*ay framework. Some performance metrics are presented to demonstrate the suitability of this platform for running High Throughput Computing applications.

1 Introduction

Impact cratering is an important geological process of special interest in Astrobiology that affects the surface of nearly all celestial bodies such as planets and satellites. The detailed morphologies of impact craters, which will not be described in this work, show many variations from small craters to craters with central peaks. Furthermore, a water layer at the target influences lithology and morphology of the resultant crater. That is the reason why marine-target impact cratering simulation plays an important role in studies which involve hypothetical martian seas [1].

In this study, we analyze the threshold diameter for cratering the seafloor of an hypothetical martian sea during the first steps of an impact. The numerical simulation of this process involves the execution of a high number of tasks, since the search space of input parameter values includes the projectile diameter, the water depth and the impactor velocity. Furthermore, the execution time of each task is not uniform because of the different numerical properties of each

* This research was supported by Ministerio de Ciencia y Tecnología, through the research grant TIC 2003-01321 and 2002-12422-E, and by Instituto Nacional de Técnica Aeroespacial "Esteban Terradas" (INTA) – Centro de Astrobiología.

R. Meersman et al. (Eds.): OTM Workshops 2004, LNCS 3292, pp. 50–58, 2004.

experimental configuration. Grid technology is a promising platform to execute this kind of applications, since it provides the end user with a performance much higher than that achievable on any single organization. However, the scheduling of each task on a Grid involves challenging issues due to the unpredictable and heterogeneous behavior of both the Grid and the numerical code. For this reason, the application will be executed on the Grid through the Grid*W*ay framework, that provides the adaptive and fault tolerance functionality required to harness Grid resources.

Results of these analysis can be used to develop a search criteria for future investigations, including techniques that will be used in future Mars exploration missions to detect buried geological structures using ground penetrating radar surveys, as the ones included in the ESA Mars Express and planned for NASA 2005 missions. The discovery of marine-target impact craters on Mars would also help to address the ongoing debate of whether large water bodies occupied the northern plains of Mars and help to constrain future paleoclimatic reconstructions [1].

We describe the target application and the Grid*W*ay framework in Sections 2 and 3, respectively. In Section 5, we present some experimental results, obtained in our research testbed, summed up in Section 4. Finally, we end with some conclusions.

2 Simulation of Impact Cratering

The impact process can be described as a transfer of energy process. The initial kinetic energy of the projectile does work on the target to create a hole –the crater– as well as heating the material of both projectile and target. We focus our attention in high-velocity impacts which can be separated into several stages dominated by a specific set of major physical and mechanical processes.

The main stages are contact and shock compression, transient cavity growth by crater material ejection, and finally, transient cavity modification. Impact cratering begins with a sufficient compression of target and projectile materials. The energy released by deceleration of the projectile results in the formation of shock waves and its propagation away from the impact point. The projectile's initial kinetic energy redistributes into kinetic and internal energy of all colliding material. The internal energy heats both the projectile and target and, for strong enough shock waves, this may result in melting and vaporization of material near the impact zone.

To describe the impact process we solve equations of motion for compressible media using a hydrocode. The standard set of equations of motion expresses 3 basic law: mass, momentum, and energy conservation. It must be combined with the equations of state (EOS), a system of relationships which allow us to describe the thermodynamic state for materials of interest. In its basic form, an EOS should define what is the pressure in the material at a given density and temperature. In an extended form, an EOS should define also the phase state of the material (melting, vapor, dissociation, ionization process) as well as all

useful derivatives of basic parameters and transport properties (sound speed, heat capacity, heat conductivity, etc.).

Numerical simulations use the Eulerian mode of SALE-B, a 2D hydrocode modified by Boris Ivanov based on SALES-2 [2]. The original hydrocode, Simplified Arbitrary Lagrangian-Eulerian (SALE), permits to study the fluid-dynamics of 2D viscous fluid flows at all speeds, from the incompressible limit to highly supersonic, with an implicit treatment of the pressure equation, and a mesh rezoning philosophy. The PDE solved are the Navier-Stokes equations. The fluid pressure is determined from an EOS and supplemented with an artificial viscous pressure for the computation of shock waves. SALES-2 can also model elastic and plastic deformation and tensile failure.

The code is based on finite difference approximations to the differential equations of motion. The algorithm used in the code is separated into four sections: problem set-up, cycle initialization, Lagrangian computation, and cycle completion.

3 The Grid *W*ay Framework

The Globus toolkit [3] supports the submission of applications to remote hosts by providing resource discovery, resource monitoring, resource allocation, and job control services. However, the user is responsible for manually performing all the submission stages in order to achieve any functionality: selection, preparation, submission, monitoring, migration and termination [4,5]. Hence, the development of applications for the Grid continues requiring a high level of expertise due to its complex nature. Moreover, Grid resources are also difficult to efficiently harness due to their heterogeneous and dynamic nature. In a previous work [6], we have presented a new Globus experimental framework that allows an easier and more efficient execution of jobs on a dynamic Grid environment in a "submit and forget" fashion. The Grid *W*ay framework provides resource selection, job scheduling, reliable job execution, and automatic job migration to allow a robust and efficient execution of jobs in dynamic and heterogeneous Grid environments based on the Globus toolkit.

Grid *W*ay achieves an efficient execution of Parameter Sweep Applications (PSA) in Globus-based Grids by combining: *adaptive scheduling*, *adaptive execution*, and *reuse of common files* [7]. In fact, one of the main characteristics of the Grid *W*ay framework is the combination of adaptive techniques for both the scheduling and execution [8] of Grid jobs:

- *Adaptive scheduling*: Reliable schedules can only be issued considering the dynamic characteristics of the available Grid resources [9,10,11]. In general, adaptive scheduling can consider factors such as availability, performance, load or proximity, which must be properly scaled according to the application needs and preferences. Grid *W*ay periodically gathers information from the Grid to adaptively schedule pending tasks according to the application demands and Grid resource status [7]. Grid *W*ay periodically gathers information from the Grid and from the running or completed jobs to adaptively

schedule pending tasks according to the application demands and Grid resource status [7].
- *Adaptive execution*: In order to obtain a reasonable degree of both application performance and fault tolerance, a job must be able to migrate among the Grid resources adapting itself to events dynamically generated by both the Grid and the running application [12,13,14]. Grid*W*ay evaluates each *rescheduling event* to decide if a migration is feasible and worthwhile [6]. Some reasons, like job cancellation or resource failure, make Grid*W*ay immediately start a *migration* process. Other reasons, like "better" resource discovery, make Grid*W*ay start a *migration* process only if the new selected resource presents a higher enough rank. In this case, the time to finalize and the migration cost are also considered [6,15].
- *Reuse of common files*: Efficient execution of PSAs can only be achieved by re-using shared files between tasks [11,16]. This is specially important not only to reduce the file transfer overhead, but also to prevent the saturation of the file server where these files are stored, which can occur in large-scale PSAs. Reuse of common files between tasks simultaneously submitted to the same resource is achieved by storing some files declared as *shared* in the Globus GASS cache [7].

In the case of adaptive execution, the following rescheduling events, which can lead to a job migration, are considered in Grid*W*ay [6,8]:

- Grid-initiated rescheduling events:
  - "Better" resource discovery [15].
  - Job cancellation or suspension.
  - Remote resource or network failure.
- Application-initiated rescheduling events:
  - Performance degradation.
  - Change in the application demands.

In this work, we do not take advantage of all the Grid*W*ay features for adaptive execution, since they are not supported by the application. In order to fully support adaptive execution, the application must provide a set of restart files to resume execution from a previously saved checkpoint. Moreover, the application could optionally provide a performance profile to detect performance degradations in terms of application intrinsic metrics, and it could also dynamically change its requirement and ranking expressions to guide its own scheduling process [8]. Therefore, we only consider adaptive execution to provide fault tolerance by restarting the execution from the beginning.

4 The Research Testbed

Table 1 shows the characteristics of the machines in the research testbed, based on Globus Toolkit 2.4 [3]. The testbed joins resources from 5 sites, all of them connected by RedIRIS, the spanish research and education network. The geographical distribution of sites is shown on Figure 1. This organization results in

Table 1. Characteristics of the machines in the research testbed.

Name	Site	Nodes	Model	Speed	Memory	OS	Job mgr.
hydrus	DACYA-UCM	1	Intel P4	2.5GHz	512MB	Linux 2.4	fork
cygnus	DACYA-UCM	1	Intel P4	2.5GHz	512MB	Linux 2.4	fork
cepheus	DACYA-UCM	1	Intel PIII	600MHz	256MB	Linux 2.4	fork
aquila	DACYA-UCM	1	Intel PIII	700MHz	128MB	Linux 2.4	fork
babieca	LCASAT-CAB	5	Alpha DS10	450MHz	256MB	Linux 2.2	PBS
platon	REDIRIS	2	Intel PIII	1.4GHz	512MB	Linux 2.4	fork
heraclito	REDIRIS	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

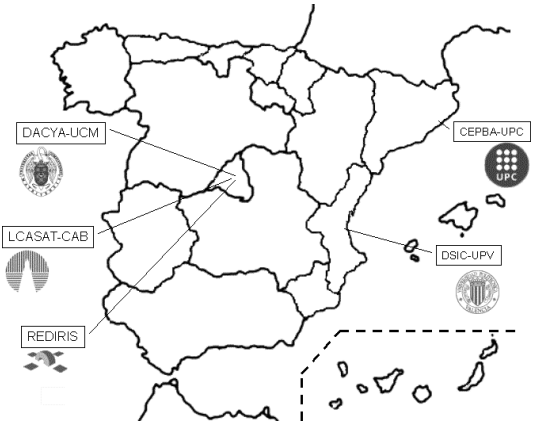

Fig. 1. Geographical distribution of sites in Spain.

a highly heterogeneous testbed, since it presents several architectures, processor speeds, job managers, network links... In the following experiments, cepheus is used as client, and it holds all the input files and receives the simulation results.

5 Experimental Results

We deal in this study with vertical impacts, as they reduce to 2D problems using the radial symmetry. All simulations were conduced with spherical projectiles. We run a set of simulations with a low-resolution computational mesh in a Grid environment. The non-uniform computational mesh of the coarse simulations consists of 151 nodes in horizontal direction and 231 nodes in vertical direction and the total nodes describes half of the crater domain because of axial symmetry. The mesh size progressively increases outwards from the center with a 1.05 coefficient to have a larger spatial domain. The central cell region around the impact point where damage is greater, more extended than the crater area, is a

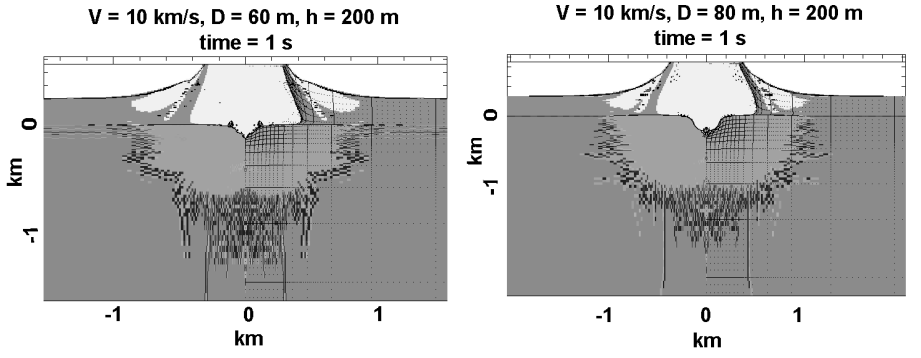

Fig. 2. Timeframes of the opening cavities at 1 second time using the 60 m impactor (left-hand chart), and the 80 m impactor (right-hand chart) with 200 m water depth and a velocity of 10 Km/s for the impactor.

regular mesh 80 nodes resolution in both x and y direction, and also describes half of the damaged zone. We use a resolution of 10 nodes to describe the radial projectile.

For a fixed water depth, we used 8 cases of projectile diameter in the range of 60 m to 1 Km, and 3 cases of impactor velocity: 10, 20 and 30 Km/s. Calculations were performed for 3 cases of water depth: 100, 200 and 400 m. Once fixed the projectile velocity and the water depth of the hypothetical ocean, we search to determine the range for the critical diameter of the projectile which can crater the seafloor. Therefore, in this study we have to compute 72 cases. The execution on a Grid environment permits to compute the 72 cases in a fast way and to obtain the diameter range of interest.

Figure 2 shows the timeframes of the opening cavities at 1 second time using the 60 and the 80 m impactor with 200 m water depth and a velocity of 10 Km/s for the impactor. The shape difference between the 60 m case and the 80 m case illustrates the water effect. Due to the water layer, in that case, the impactor diameter has to be larger than 80 m to crater the seafloor.

The execution time for each task is not uniform, since the convergence of the iterative algorithm strongly depends on input parameters, besides the differences produced by the heterogeneous testbed resources. Moreover, there is an additional variance generated by the changing resource load and availability. Therefore, adaptive scheduling is crucial for this application. Figure 3 shows the dynamic throughput, in terms of mean time per job, as the experiment evolves. The achieved throughput is 3.87 minutes per job, or likewise, 15.5 jobs per hour. Total experiment time was 4.64 hours. Table 2 shows the schedule performed by Grid$W$ay, in terms of number of jobs allocated to each resource and site. Most of the allocated jobs were successfully executed, but others failed and had to be dynamically rescheduled.

Given the results on Table 2, we can calculate the fault rate for each resource or site. It is clear that all resources (sites), but babieca and ramses (LCASAT-

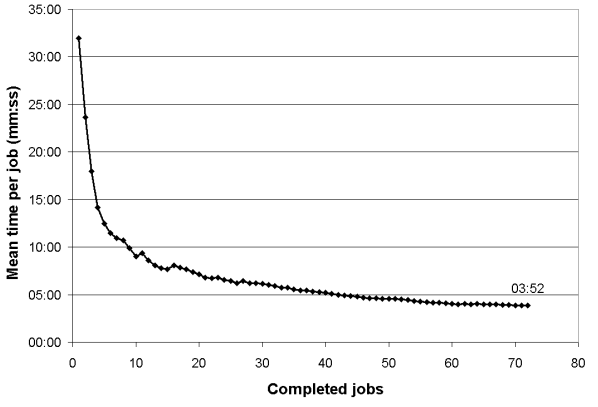

Fig. 3. Dynamic throughput, in terms of mean time per job.

Table 2. Schedule performed by Grid*W*ay, in terms of jobs allocated to each resource (left-hand table) and site (right-hand table).

Resource	Allocated	Done	Failed	Ratio
aquila	4	4	0	0%
babieca	24	18	6	25%
cygnus	7	7	0	0%
heraclito	2	2	0	0%
hydrus	8	8	0	0%
khafre	12	12	0	0%
platon	5	5	0	0%
ramses	29	16	13	45%
Total	91	72	19	21%

Site	Allocated	Done	Failed	Ratio
CEPBA-UPC	12	12	0	0%
DACYA-UCM	19	19	0	0%
DSIC-UPV	29	16	13	45%
LCASAT-CAB	24	18	6	25%
REDIRIS	7	7	0	0%
Total	91	72	19	21%

CAB and DSIC-UPV), have a fault rate of 0%, and the two failing resources (sites) have a fault rate of 25% and 45%, respectively, which supposes an overall fault rate of 21%. These failures are mainly due to a known Globus problem (bug id 950) related to NFS file systems and the PBS job manager. This problem is mitigated, but not avoided, on babieca, where a patch related to this problem was applied.

Figure 4 shows the achieved throughput, also in terms of mean time per job, by each site and by the whole testbed for the above schedule. In the right axis, the distributed or Grid speed-up (i. e. the performance gain obtained by each site) is also shown. We think that is a valuable metric for resource users and owners on each site to realize the benefits of Grid Computing, since results like this can help to curb their selfishness [5].

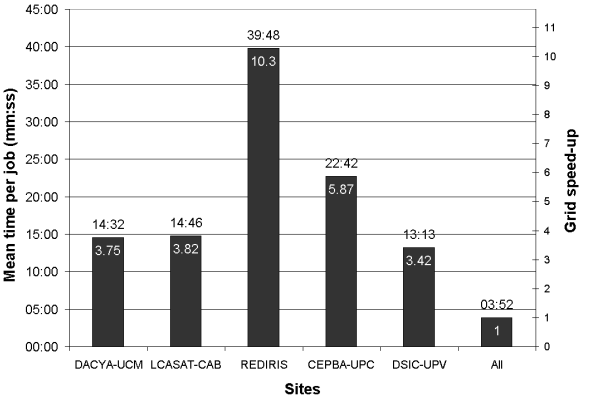

Fig. 4. Throughput, in terms of mean time per job (left-hand axis and values on top of columns) and Grid speed-up (right-hand axis and values inside columns), for each site and for the whole testbed (rightmost column, labelled as "All").

6 Conclusions

Studies about marine-target impact cratering will help to develop a search criteria for future investigations and exploration missions to Mars and, if they are successful in their hunt, they would also help to address the ongoing debate of whether large water bodies existed on Mars and, therefore, they would help to constrain future paleoclimatic reconstructions.

This kind of studies requires an huge amount of computing power that is not usually available in a single organization. Grid technology, and Globus in particular, allows the federation of resources from different organizations, with respect for each site autonomy, to help in the construction of virtual organizations. However, efficient 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. That has been demonstrated by performance metrics like the Grid speed-up, which is a valuable metric for resource users and owners to realize the benefits of sharing resources over the Grid.

Acknowledgement. We would like to thank all the research centers that generously contribute resources to the experimental testbed. They are the European Center for Parallelism of Barcelona (CEPBA) in the Technical University of Catalonia (UPC), the Department of Computer Architecture and Automatics (DACyA) in the Complutense University of Madrid (UCM), the Department of Information Systems and Computation (DSIC) in the Polytechnic University of Valencia (UPV), the Laboratory of Advanced Computing, Simulation and Telematic Applications (LCASAT) in the Center for Astrobiology (CAB), and the Spanish Research and Education Network (RedIRIS). All of them are part of the Spanish Thematic Network on Grid Middleware.

We would like to also thank Jens Ormö and Jesús Martínez-Frías, in the Planetary Geology Laboratory at CAB, for their help in understanding the impact cratering simulations.

References

1. Ormö, J., Dohm, J.M., Ferris, J.C., Lepinette, A., Fairén, A.: Marine-Target Craters on Mars? An Assessment Study. Meteoritics & Planetary Science **39** (2004) 333–346
2. Gareth, S.C., Melosh, H.J.: SALES 2: A Multi-Material Extension to SALE Hydrocode with Improved Equation of State and Constitutive Model. Available at http://www.lpl.arizona.edu/~gareth/publications/sales_2 (2002)
3. Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. Intl. J. Supercomputer Applications **11** (1997) 115–128
4. Schopf, J.M.: Ten Actions when Superscheduling. Technical Report GFD-I.4, Scheduling Working Group – The Global Grid Forum (2001)
5. Schopf, J.M., Nitzberg, B.: Grids: The Top Ten Questions. Scientific Programming, special issue on Grid Computing **10** (2002) 103–111
6. Huedo, E., Montero, R.S., Llorente, I.M.: A Framework for Adaptive Execution on Grids. Intl. J. Software – Practice and Experience (SPE) **34** (2004) 631–651
7. Huedo, E., Montero, R.S., Llorente, I.M.: Experiences on Adaptive Grid Scheduling of Parameter Sweep Applications. In: Proc. 12th Euromicro Conf. Parallel, Distributed and Network-based Processing (PDP2004), IEEE CS (2004) 28–33
8. Huedo, E., Montero, R.S., Llorente, I.M.: Adaptive Scheduling and Execution on Computational Grids. J. Supercomputing (2004) (in press).
9. Berman, F., et al.: Adaptive Computing on the Grid Using AppLeS. IEEE Trans. Parallel and Distributed Systems **14** (2003) 369–382
10. Buyya, R., et al.: Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computation Grid. In: Proc. 4th Intl. Conf. High Performance Computing in Asia-Pacific Region (HPC Asia). (2000)
11. Casanova, H., et al.: Heuristics for Scheduling Parameter Sweep Applications in Grid Environments. In: Proc. 9th Heterogeneous Computing Workshop. (2000)
12. Allen, G., et al.: The Cactus Worm: Experiments with Dynamic Resource Discovery and Allocation in a Grid Environment. Intl. J. High-Performance Computing Applications **15** (2001)
13. Lanfermann, G., et al.: Nomadic Migration: A New Tool for Dynamic Grid Computing. In: Proc. 10th Symp. High Performance Distributed Computing (HPDC10). (2001)
14. Vadhiyar, S., Dongarra, J.: A Performance Oriented Migration Framework for the Grid. In: Proc. 3rd Intl. Symp. Cluster Computing and the Grid (CCGrid). (2003)
15. Montero, R.S., Huedo, E., Llorente, I.M.: Grid Resource Selection for Opportunistic Job Migration. In: Proc. 9th Intl. Conf. Parallel and Distributed Computing (Euro-Par 2003). Volume 2790 of LNCS. (2003) 366–373
16. Giersch, A., Robert, Y., Vivien, F.: Scheduling Tasks Sharing Files on Heterogeneous Master-Slave Platforms. In: Proc. 12th Euromicro Conf. Parallel, Distributed and Network-based Processing (PDP 2004), IEEE CS (2004) 364–371