A Distributed Computing System for Near Earth Objects Hazard Monitoring

Pasquale Tricarico   Washington State University

Introduction

The collaboration among professional astronomers, Celestial Mechanics researchers, and amateur astronomers initially started to successfully discover and follow Near Earth Objects. This collaboration should now be extended toward the realization of a Distributed Computing system for the analysis of the astrometric data for hazard monitoring purposes, *allowing the participation of the public* in the computational effort required to monitor the hazard posed by these objects. After a brief introduction to Near Earth Objects and Distributed Computing systems, we describe how a Distributed Computing system for Near Earth Objects hazard monitoring can be developed, it's benefits and advantages over non-distributed systems. *Work is underway to realize this distributed computing system.*

Near Earth Objects

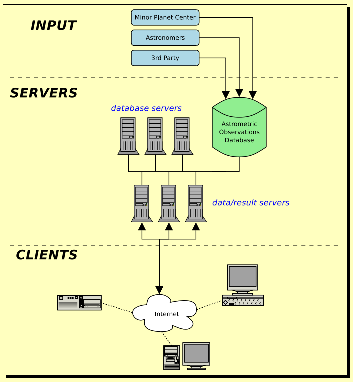
All the asteroids and comets that can have close approaches with the Earth can be considered Near Earth Objects (NEOs). The size of these objects is usually between a few meters and a few kilometers.

Since there is a *small but non-zero* chance of impact between some not-yet-discovered NEO and the Earth (Chapman & Morrison, 1994; Chesley *et al.*, 2002; Chapman, 2004; Stuart & Binzel, 2004), the detection of these objects represents the main objective of many surveys. The orbits of all the known NEOs are monitored by two independent automatic systems, *NEODyS* (Chesley & Milani, 2000) and *Sentry* (Chamberlin *et al.*, 2001). Both systems provide impact probabilities, orbit uncertainties and other useful data for each NEO.

Distributed Computing

Distributed Computing (DC) is the natural frame for the solution of numerical problems where a task can be divided into independent pieces, and whose ratio of computation to data is high (Anderson, 2004). Every work unit is sent to a different computer, while the central system collects and analyzes the results. Examples of DC systems include: the *Great Internet Mersenne Prime Search* (GIMPS, *mersenne.org*), the *Search for Extraterrestrial Intelligence* (SETI@home, *setiathome.ssl.berkeley.edu*), *Protein Folding* (Folding@home, *folding.stanford.edu*) and *Climate Prediction* (*climateprediction.net*).

Near Earth Objects *plus* Distributed Computing

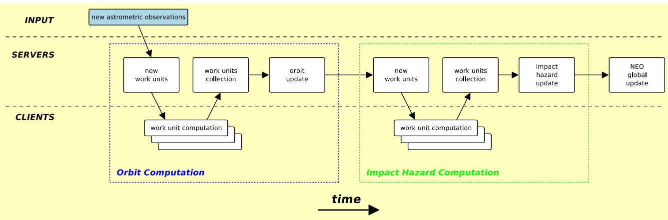


A DC system for NEOs impact hazard monitoring can be developed, with a structure like the one showed in this diagram. The three main sections of this system are the input, the servers and the clients sections. The **input** is represented by all the astronomical and radar observations of the NEOs. This data is provided mainly by the Minor Planet Center (MPC), but in special cases can be provided directly by astronomers or by 3rd parties. The **servers** section represent the core of the DC system: the servers collect astrometric data, manage the asteroids database, provide and collect work units to/from the clients, and provide any other central service needed. The **clients** perform all the numerical computations. They receive a work unit from a server, perform all the needed numerical computations, and return the results when finished. Advanced clients can show the status of the DC system, with a list of important NEOs, latest system updates, and other useful informations. The *size* of the work units is small enough so that can be transferred via Internet using dial-up lines, but is also big enough to keep the client working for a sufficient amount of time.

The NEOs+DC System Flowchart

The most important aspect of this system is the fact that *every NEO can be monitored independently from the others*. This means that every client can compute a work unit relative to a different NEO, without the need of any communication or synchronization with other clients. This point is probably the most important to justify the realization of this DC system. In a non-distributed system, this aspect cannot be fully exploited, because usually the number of CPUs available is fixed and is much smaller than the number of NEOs that needs to be updated. In a DC system, there is no limit to the number of clients that can work at the same time, and if a sufficient number is available, and if all the clients promptly start the computations needed, all the work units can be processed at the same time, i.e. all the NEOs can be updated without delay. Of course, in the DC system some extra-work is needed to generate work units, send and receive data over Internet, and also some redundancy factor has to be kept into account.

The redundancy in the number of clients processing the same work unit is necessary to cross-check the results and to avoid data corruption, wrong results due to hardware failure and other problems.
Since clients can be developed for different hardwares and operative systems, and the internals of each system are different from client to client, the numerical results of the same work units computed on different clients can be significantly different. Far from being a limit, the study of these differences allows to better understand the numerical behavior of the algorithms used, allows a deep debugging of the computer code, and permits to estimate impact probabilities in a system-independent frame.
Most DC systems are characterized by a large amount of data that needs to be analyzed or simulated. The data is continuously sent to clients, and every work unit takes at least one and up to some days to finish. The DC system we describe here is significantly different: new data is available on a daily basis, the typical work unit size is relatively small, and all the data can be analyzed by a few thousand clients in a much shorter period.



This diagram describes the series of operations needed every time new observations are available for a NEO. The two main section represent the orbit update and the impact hazard update. The first computation is needed to obtain an updated orbit for the NEO. All the observations available are sent to the clients, to determine the new orbit using different methods. When a new orbit for the NEO has been determined, the impact monitoring computations can start, and new work units are made available. We estimate that each orbit determination work unit can require an average of about 1-2 minutes of CPU time on an average modern personal computer, while the impact monitoring computations will likely need more CPU time, probably about 10 minutes. With a sufficient number of clients available, it is possible in principle to update the whole NEOs database in about 15 minutes. However, these times will strongly depend on the number of clients available, and on the *size* of the work units, and possibly some other parameters, so they can change significantly.

Public Computing

Distributed Computing is also *Public Computing*, as most of the computing power comes from the public. *"Public computing can provide more computing power than any other supercomputer, cluster, or grid, and the disparity will grow over time (Anderson, 2004)."* Usually most participants are motivated primarily by the underlying science, and in the case of NEOs hazard monitoring, motivated users can learn more about impact hazard and probability, and how to correctly read Palermo and Torino Scales. The community that eventually develops around the project can help reducing the gap between NEO science and the public.

Project Status

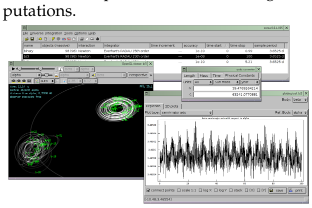
The DC system we describe here is under very active development. The main components of the system are the BOINC platform and the ORSA framework. While BOINC is mature enough and is used in many DC systems, the Celestial Mechanics and impact hazard monitoring algorithms from ORSA are still under development, and probably a few months of work are still needed. We estimate the public availability of clients not earlier than spring 2005.

The BOINC Platform

The *Berkeley Open Infrastructure for Network Computing* (BOINC, *boinc.berkeley.edu*) is a software platform for distributed computing using volunteered computer resources (Anderson, 2004). This platform allows to develop a Distributed Computing system using a flexible application framework, and provides security features, multiple servers support and fault-tolerance, system monitoring tools and support for large data. The BOINC source code is available under a public license.

The ORSA Framework

The *Orbit Reconstruction, Simulation and Analysis* (ORSA, *orsa.sourceforge.net*) project provides the framework for the development of the Celestial Mechanics algorithms needed for impact hazard monitoring computations.



The ORSA project also provides a standalone graphical application that permits to simulate any planetary system, and offers high quality graphical tools, asteroids and comets import from many different Minor Planets databases, all using fast and accurate numerical algorithms. The ORSA source code is available under the GPL license, and precompiled binaries are available for Linux, Mac OS X and Windows.

Extensions

Depending on the number of contributing users, this DC system can be extended i.e. to study different asteroid deflection methods, or to monitor impacts with other planets in the Solar System.

Conclusions

We have described a possible implementation of a Distributed Computing system for NEOs impact hazard monitoring purposes. This system allows to monitor NEOs with a very short delay between observations and impact hazard assessment, and allows the public to get involved in the NEO science. This system is completely open source, and can be easily extended to study many other Minor Planets related phenomena.

Contact

Pasquale Tricarico <*tricaric@wsu.edu*>

Acknowledgments

We gratefully acknowledge George Lake, Nathan Hearn, Sandor Molnar and Francesco Marzari for their suggestions and their interest in this work.

References

C. R. Chapman and D. Morrison, *Impacts on the Earth by asteroids and comets: assessing the hazard*, Nature, Vol. 367, p. 33-40 (1994).

S. R. Chesley, A. Milani, *An Automatic Earth-Asteroid Collision Monitoring System*, BAAS, Vol. 32, p. 862 (2000).

A. B. Chamberlin, S. R. Chesley, P. W. Chodas, J. D. Giorgini, M. S. Keesey, R. N. Wimberly, D. K. Yeomans, *Sentry: An Automated Close Approach Monitoring System for Near-Earth Objects*, BAAS, Vol. 33, p. 1116 (2001).

S. R. Chesley, P. W. Chodas, A. Milani, G. B. Valsecchi, D. K. Yeomans, *Quantifying the Risk Posed by Potential Earth Impacts*, ICARUS, Vol. 159, p. 423-432 (2002).

D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky and D. Werthimer, *SETI@home: An Experiment in Public-Resource Computing*, Communications of the ACM, Vol. 45, No. 11 (2002).

C. R. Chapman, *The hazard of near-Earth asteroid impacts on earth*, Earth and Planetary Science Letters, Vol. 222, p. 1-15 (2004).

J. S. Stuart and R. P. Binzel, *Bias-corrected population, size distribution, and impact hazard for the near-Earth objects*, ICARUS, Vol. 170, p. 295-311 (2004).

D. P. Anderson, *Public Computing: Reconnecting People to Science*, presented at the Conference on Shared Knowledge and the Web, Residencia de Estudiantes, Madrid, Spain (2004).