The Challenge of Volunteer Computing With Lengthy Climate Model Simulations

Carl Christensen, Tolu Aina and David Stainforth
Department of Atmospheric Physics
University of Oxford
Clarendon Laboratory
Oxford OX1 3PU
United Kingdom
(carlc,tolu,das)@atm.ox.ac.uk

Abstract

This paper describes the issues confronted by the climateprediction.net project in creating a volunteer computing project using a large legacy climate model application. This application typically takes from weeks to months to complete one simulation, and has large memory and disk usage requirements. We describe issues in porting the climate model to a single-processor PC platform, checkpointing, computer resources and workunit size for a simulation, and the volunteer computing infrastructures used (a project-specific system and BOINC). We also describe the methods used to obtain and retain users, and examine the retention/attrition rate of users running the lengthy modelling simulations.

1 Introduction

The success of the SETI@home [5] project has made volunteer computing a new "killer app." It is popular with the global community of participants who volunteer their computer time for various projects of interest. And it is an increasingly popular option for scientists with insufficient computational resources for their research. As far back as 1999, climate scientists wondered if it would be possible to use the power of volunteer computing in running full-scale climate models [3]. This paper provides an overview of the issues involved in making this a reality via the climate*prediction*.net (CPDN) project. We also discuss how this project differs from other volunteer computing applications. We hope to provide guidance for future modelling initiatives or other applications with long-term workunits using the volunteer computing paradigm.

The climate*prediction*.net project uses home PCs to run an atmosphere-ocean general circulation model (AOGCM) — a type of program that is generally run on supercomputers and other parallel machines such as Beowulf clusters [7]. We will describe the development and deployment of the project from the original design [9, 10] and through the move to the Berkeley Open Infrastructure for Network Computing (BOINC) [1, 4].

Section II presents the issues faced in porting the large legacy codebase to a single-processor personal computer. Section III presents the issues faced by a volunteer computing application that has lengthy workunits. Section IV provides a look at the volunteer computing infrastructures utilized by climate*prediction*.net. The progress of the actual user base of climate*prediction*.net and their retention/attrition rates is given in Section V. Finally, future plans are presented in Section VI, with concluding remarks in Section VII.

2 Porting the Climate Model to a PC

climate*prediction*.net uses the UK MetOffice "Unified Model" (UM) [6, 13]. This model has been tested and peer-reviewed in the scientific literature, and is familiar to many climate scientists, allowing them to easily study the database of completed runs. The use of the UM model gives the public the opportunity to run a "state of the art" climate model, and introduce them to such issues as global warming, climate research and climate modelling.

The UM is comprised of approximately one million lines of Fortran code (40 megabytes of source code) across 600 files. The size and structure of this large legacy Fortran codebase provided numerous problems to project software developers. Much work was done in both porting and integrating the model into the object-oriented (C++) controlling (monitoring) program and volunteer computing system.

There are also many security issues for various stakeholders on a volunteer computing project and especially a modelling simulation such as climate*prediction*.net — these are presented in a separate paper [12].

2.1 Porting and Compiler Issues

The "Hadley Centre Slab Model 3" (HadSM3) was selected as the most promising version of the UM for the first phase of the project, as it required fewer computer resources than versions with more complex oceans and higher resolutions. The model divides the world into "grid boxes" with a resolution of 2.75 by 3.75 degrees per grid box (73 latitudinal and 96 longitudinal "slices"). This is a fairly typical model resolution, and the 45-year simulations required runs on a single processor home PC (i.e. a Pentium IV/2GHz machine) in approximately 4-6 weeks.

Initial development was for the Microsoft Windows operating system (the operating system of most volunteer computing users). As climate*prediction*.net moved to the BOINC platform in 2004, the application was ported to Linux and Apple Macintosh OS X. The task of porting the UM Fortran code (F90) to Windows involved a port to Linux as an intermediate stage as the UM had previously been run on single-processor Unix systems.

Many tests were then done of the ported climate model using setups that could be compared with equivalent runs on a Cray supercomputer and Beowulf clusters. Eventually it could be demonstrated that the use of the climate model on a large ensemble of PCs via volunteer computing was a viable method ([11]).

2.2 Checkpointing

Scientific applications being ported to the volunteer computing paradigm must *checkpoint* time-consuming tasks (e.g. greater than an hour of run-time). This enables a restart of the task with little loss of previously computed work. Many scientific applications were meant to be run continuously from "start to finish," with job submission by researchers who patiently await the results on a time-shared system, and who do not interrupt the task. Therefore, scientific programs often have no checkpointing capability.

For a volunteer computing app this is not desirable, as user intervention, system crashes, and other factors may require a task to be paused, stopped, or removed from memory, and later restarted. Fortunately, checkpointing is available for the climate model used in climate*prediction*.net. The checkpoint of the climate model requires approximately 20MB of file I/O, which is larger than typical volunteer computing apps that may need to simply write out a few kilobytes in state files. Because of this larger file I/O requirement, checkpointing on climate*prediction*.net is done every 144 timesteps (3 simulated days of the model). This occurs approximately every 15 minutes on a Pentium IV/2GHz machine, and seemed to be the best compromise of file I/O for user restarts. So at most 15 minutes of CPU time is lost when the user stops the model or shuts down the computer.

2.3 Operational Requirements

Most volunteer computing projects such as SETI@home and Einstein@home break up workunits into small datasets for ease of download by the user, as well as to give a workunit that can typically be completed in a matter of hours by a user's machine. As mentioned previously, the climate model does have the facility to do checkpoints via "restart dumps" at a certain number of model timesteps. These "restart dumps" could theoretically be returned as frequently as other volunteer computing workunits (with perhaps a completed workunit being a model-year simulated, as opposed to the full 45 model-years of a run). However this would require the frequent (i.e. once or twice daily) downloading and uploading of the 10 megabyte checkpointed "restart dumps." This would preclude participation in the project by many modem users or those with limited bandwidth. Also, it was decided early on in the project that for educational (i.e. "user outreach") purposes it is more effective for the user to follow the "evolution" of their particular model version from start to finish.

Table 1 shows the requirements of running the climate model on a PC, with a comparison to resource usage for other volunteer computing applications. One can see that climate*prediction*.net has much larger requirements for computer hardware resources and CPU time per workunit as compared with other distributed computing projects — two orders of magnitude for disk and CPU time. *Table 2* shows CPU time execution for various volunteer computing (BOINC) projects run on an Intel Xeon/3.4GHz machine. Note the CPU time of three weeks to finish one HadSM3 workunit, even for this very fast machine (by July '05 standards). Typically, a single run of a climate model takes about a month for a "later generation" PC, but some users have returned results more than a year after the initial download!

A volunteer computing application needs to consider the bandwidth requirements and availability to both the user community and the project servers. Although broadband connections (DSL and cable) are becoming increasingly common, many users still rely on 28K and 56K modem connections for Internet access. Therefore, a volunteer computing application would hopefully only require a few megabytes for initial software download, and possibly less of a download required for each workunit.

On climate*prediction*.net the installation package (i.e.

compiled model and other executables and data files) is a 10MB download. This is slightly more than other volunteer computing projects, but is still feasible to download on a modem (within an hour). The climate model produces hundreds of megabytes during a run. Out of this large dataset, only about 8MB of summary information (e.g. multi-year means of important fields such as temperature, precipitation, etc) is compressed and returned to the project. Given that it typically takes 3-6 weeks to complete a simulation, an 8MB upload should not present a problem for modem users (at worst an hour upload every month). Additional workunits are only a 20KB compressed download of Fortran namelist files.

3 Infrastructure for volunteer computing

With the climate model ported, it was then possible to look more into the client graphical user interface (GUI) as well as the client and server-side requirements for supporting the volunteer computing project. Although porting the million lines of legacy Fortran code to a PC was a daunting task, it took far less time than developing the "wrapper" code around the model to make it a volunteer computing application.

The infrastructure necessary for volunteer computing is essentially a "vertical application" that must handle the distribution of software and workunits to users worldwide, the web pages they will see for their work statistics, the network communications between the users and project servers, etc. Since a suitable, generic framework did not exist at the time, the climate*prediction*.net project had to build this infrastructure in addition to the model porting and testing. The porting of the model took a few months of development time; the volunteer computing infrastructure required about two years (for two full-time software engineers).

Now, the BOINC framework encapsulates all aspects of the "vertical application" required for volunteer computing. This includes the client-side software APIs that can be added to the scientific code, server-side workunit delivery and result uploading code, the database (mySQL), and the user and project maintenance web pages. Climate*prediction*.net's migration to BOINC has enabled the projects' two full-time software engineers to concentrate their time on climate modelling and scientific research issues. This enables the production of a more stable volunteer computing application, with better crash recovery and fault tolerance of the climate model.

As an example of the benefits of the move to the BOINC framework — the first phase of climate*prediction*.net was for Windows only and took approximately 2.5 years of development (because of the need for the "vertical application" for volunteer computing). Since using BOINC in early 2004, it has only taken six months to incorporate the same HadSM3 model for Windows as well as port the model to the Linux and Apple Mac OS X platforms. Another benefit of using the BOINC framework is that one can share a community of users of other BOINC projects, and when one project is down for maintenance, another project can benefit (even temporarily) from an increase of users "crunching" their project.

4 Volunteer Computing User Base

In addition to the scientific and computing issues presented in creating a volunteer computing application, the process of obtaining and retaining users is very important. This is intensified when the application is a very resource-intensive system with lengthy workunits such as climate*prediction*.net. It is hard to predict how many users a project will get, and what the retention/attrition rate will be.

The system was designed to support two million users, each completing a single model run. This was an "optimistic" number based on the SETI@home project having about five million total users. It should also be noted that at the time of the design, a single model run was estimated to take about 3-6 months. Thanks to "Moore's Law" this has decreased; a typical run now takes 4-6 weeks. The actual total user base of climate*prediction*.net (on the original and BOINC infrastructures) is 150,000, with about 30,000 active users at any one time. This user base has completed just over 100,000 full (45 model-years) runs (as of July '05) – which represents 4.5 million simulated model-years.

4.1 Obtaining Users

Volunteer computing on the scale of the SETI@home project has much potential — but getting the word out to the public remains a great challenge. As in the 'dot.com' business world, great ideas and deserving enterprises may fail because they don't get the word of mouth recognition (or forwarded emails) that help make others succeed. The best hope for volunteer computing projects is that the project itself has a great public appeal. Marketing and public relations can help, as well as publications of news and results in the scientific literature, and, if possible, major media outlets.

On the climate*prediction*.net project, "launch events" for the original and BOINC release of the project software were used to help generate media interest. The publication of the first results in *Nature* (and the resulting media interest) in January of 2005 also generated a large increase in users [8]. For BOINC projects, mention on the main BOINC website [1] or other volunteer computing project websites or user websites can be a big help in attracting new users.

Projects that use the BOINC platform can effectively share the user base, and projects can co-exist in the BOINC client where the user has full control over what percentage of CPU time to give to a particular project. And even for users dedicated to one project, occasionally the project website will be down, or the project will be offline for maintenance, which could lead the user to try another project. The "resource sharing" provided by BOINC seems to help the climate*prediction*.net project, as users can see the longer-term benefits of contributing to many projects (and gain the credit for helping each project).

4.2 Retaining Users

The climate*prediction*.net client asks much of a participant as far as the amount of disk space, CPU time, and memory used. The client is designed to run while the PC is idle and behaves well even when the user works with other programs, as it is set to run at the lowest 'idle' priority level. However, the resource demands and length of time to finish a run seems to lead to a high attrition rate. So how does one keep a user on a longer-term volunteer computing project?

4.2.1 Graphics

On climate*prediction*.net it was decided to make an impressive 3-D OpenGL visualisation package to show model output to the user. Most volunteer computing projects have a graphic component to hold the user's interest, and these graphics can often be used in the form of a screensaver, which can perhaps "catch the eye" of another potential user. However, often the scientific problems are hard to visualize for a layperson (i.e. radio telescope or LIGO data, protein folding etc.). Conversely, a layperson can often appreciate the issue of climate change and weather, And it's relatively easy to visualize a climate model by plotting the grid points and various fields (temperature, precipitation, cloud cover, etc) on a 2-D world map or 3-D world sphere.

The climate*prediction*.net graphics (implemented using C++ and OpenGL) show the user's currently running model and the user's 'world' full of clouds, snow, and sea ice. Different views show temperature, precipitation, and pressure. Thus they can follow their 'world' as it evolves through time. For more advanced users, an add-on package written in IDL is available that allows the user to analyze the climate model data files on their computer. There is even a separate graphics package developed independently by an enthusiastic participant of the project. It is hoped that by the availability of such graphics add-ons, the user's interest will be retained through to the completion of at least one model simulation.

4.2.2 Credits

One facet of user retention that was very important to the SETI@home project (and ultimately all volunteer computing projects) was in the competitive nature of participants as they get "credits" for completing workunits. Many users enjoy competing with each other in terms of number of workunits finished and speed of computation of a workunit. Teams of computer users form and compete with one another. "Power users" show off the speed of their overclocked and supercooled machine. climate*prediction*.net was late in appreciating this aspect of volunteer computing projects; presumably thinking that the science and educational aspects, as well as the graphics, were sufficient to retain users' interest. In reality, the user credits are still a very important part of volunteer computing projects.

4.2.3 'Trickles' of Data For User Feedback

Users typically like to see their credits accrue on a daily basis, which can cause a problem for a project that has long workunits. To get around this problem, a mechanism called the 'trickle' was put into the climate*prediction*.net software. This is basically a 'ping' from the client to the central server so that the project knows which users are active, how much time they've spent running climate*prediction*.net, and where they are at in the model run. At certain intervals (i.e. the end of one of the three phases) the 'trickle' returns some diagnostic data of the model run that can be reported back to the user via an online graph on their user page. This data can also be used by scientists to see which runs are unstable, and which runs will be finished (uploaded) soon.

Without the 'trickle' mechanism, it would have taken one to three months before a user received their computational credits, saw how their team was doing, or viewed their results online. That would presumably make for a dull online experience for a user, and lead to their losing interest in the project. There was initially a fear that the 'trickle' would lead to a self-inflicted "denial of service" attack, but this has not happened. A typical 'trickle' is only about 300 bytes sent as XML via an HTTP POST, so with 100,000 users "trickling" per day (it's usually a once or twice-daily update from the client) that is only 30-60MB/day of network traffic. The 'trickle' mechanism has been ported over to BOINC for use by other projects with long workunits.

4.2.4 Interactions With Users

Most volunteer computing projects have an online message board for interacting with users, and it is apparent that many enjoy the social aspect of a project rather than just donating their computer time. The climate*prediction*.net message boards can at times be a lively debating center for topics ranging from computer hardware purchases to global warming. Users also like to hear news from the project and see participation by the scientists on the message boards — or at the very least get news updates of the project such as papers published by the project team. These forums also provide a place to assist newcomers to the project, or others seeking assistance with problems in running the project.

While the value of these interactions with volunteer computing users cannot be denied, it does take time away for members of the project to respond to questions and queries for assistance. A nice feature that seems to occur with the message boards is that many users (as they gain more experience with the project's computing or scientific aspects) are happy to act as 'gurus' or 'moderators' to the larger community.

4.3 User Retention/Attrition Rate

As a "case study" of the efficacy of various methods used to obtain and retain users, it may be useful to follow the success/failure rate of new users who signed up to the project. The two 'launch' periods were used to track the group of users signing up the week of the launch. The original climate*prediction*.net project was launched in September of 2003 at the Science Museum in London, and the climate*prediction*.net/BOINC project was launched in late August of 2004 at Cal-Berkeley. *Table III* show the progress of these two groups of new users and their machines.

One can see from Table III (and from Figure 1 as well) that the launch publicity gained quite a large number of users and machines who signed on to the project. Unfortunately, many of these users (and their machines) did not make it to even the first 'trickle' point (1.4% of model completion). This first 'trickle' point should have been reached (on a typical Windows PC of a Pentium4/2GHz) in about 12 hours of CPU time of the model. In that amount of time, most volunteer computing projects would have finished one or more workunits (i.e. two SETI@Home or one Einstein@Home completed calculation). From discussions with the user community, it appears that the climate model is very sensitive to the heterogeneous configurations that one encounters on a volunteer computing project. The myriad Intel and AMD Windows and Linux PCs possible (hardware configurations, software drivers, operating system versions), and the limited resources of the project to test all configurations, means that many computers are unable to get far into the model.

One can conclude from Table III, however, that using the BOINC framework leads to a much higher completion rate of models finished, and a higher retention rate especially in the long term, with about a third of users and machines still active after six months. The original climate*prediction*.net system has a much higher dropout rate.

Refer to Fig. 1 for a chart of user and host machine attrition throughout each 'trickle' of a full model run. Note the sharp decrease before the first timestep (mentioned above), and the slow gradual decrease throughout the lifecycle of a full run. However the slope decreases even shortly into the run (say 12.5-25%), so it seems the initial period (from signup to "first trickle") is critical. Here again, it can be seen that the BOINC users and machines have a lower attrition rate and tend to make it through the complete model run.

We hope that with future development and testing of the models on various systems, as well as making installation instructions easier, more users will survive this critical "first day" Fig. 2 shows the number of new and active machines for the original and BOINC versions of climate*prediction*.net. Even with a lower number of BOINC users starting at the launch, it can be seen that the steady increase of BOINC users soon makes for a larger pool of active users than with the original climate*prediction*.net project. It appears that as BOINC users on other projects find a new project of interest to attach to, they tend to stay with the project unless repeated errors force them to give up on the project.

The attrition rates for the lengthy runs of the climate model seem disappointing; however, they show that overall for every user who signs up, there is close to one complete model finished. This is due to many eager users who continue to run more workunits after finishing their first, thereby 'making up' for the users and machines lost. And even with this large attrition rate, the effective throughput of the project (with 30K active hosts, 100K finished 45-year runs) is equivalent to the capacity of two "Japanese Earth Simulators" [2]. This is based on experiences of running the UM climate model at the Earth Simulator facility, which suggests that it would complete approximately 5000 45-year runs per month (225K model-years). In 10 months since the CPDN/BOINC launch the project has simulated 100K full runs (4.5 million model-years).

5 Future Plans

Following on the success of the migration to the BOINC framework, the project plans (by January 2006) to release a version of the HadSM3 model with additional code to simulate a sulphur cycle, a high-resolution atmospheric model (HadAM3), and a fully dynamic coupled atmosphere/ocean model (HadCM3). These models will require increasingly more computing resources to run, and we hope that the general public will continue to upgrade their machines! If one looks back to the original climate*prediction*.net proposals and papers on running a full-scale climate model on a PC, the typical PC of the day (say 1999 to 2001) was far behind what climate*prediction*.net wanted to achieve. Thanks to the development of the BOINC framework, the enthusiasm of users around the world contributing their CPU time, and "Moore's Law" — it is now a reality.

6 Conclusion

This paper has presented various issues with running lengthy workunit and large, complex, legacy applications within the volunteer computing paradigm. While the issues presented pertain to climate modelling and volunteer computing via the climate*prediction*.net project, we believe that they will apply to other current or potential projects with lengthy workunits or a large "footprint" of computing resource requirements. These applications may use volunteer computing effectively if they have enough "mass appeal" to obtain (and retain) a large number of users. Or they can be used as a distributed computing application within a corporation, Intranet, or worldwide-distributed research group (especially in the case of proprietary code or applications). As the development of the "vertical application" for a volunteer computing project can equal or (most likely) exceed the time required for porting, the use of a proven volunteer computing infrastructure such as BOINC would be a great help to get the project released. We invite you to visit the climate*prediction*.net website for further information and to perhaps start your own run of a climate model.

Acknowledgment

We would like to thank David Anderson of the SETI@home and BOINC projects, all of the BOINC developers and contributors, and of course all of the users throughout the world who contribute their time and computers on the climate*prediction*.net project.

References

[1] URL: `http://boinc.berkeley.edu`.

[2] URL: `http://www.es.jamstec.go.jp/`.

[3] M. R. Allen. Do-it-yourself climate prediction. *Nature*, 401:642, 1999.

[4] D. P. Anderson. BOINC: A System for Public-Resource Computing and Storage. *5th IEEE/ACM International Workshop on Grid Computing*, 8 November, 2004.

[5] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. SETI@home: An Experiment in Public-Resource Computing. *Communications of the ACM*, V. 45, No. 11:56–61, November 2002.

[6] M. Cullen. The unified forecast/climate model. *Meteor. Mag.*, 122:81–94, 1993.

[7] J. A. Hansen, M. R. Allen, D. A. Stainforth, A. Heaps, and P. Stott. Casino-21: Climate simulation of the 21st century. *World Resource Review 13*, 2001.

[8] D. A. Stainforth, T. Aina, C. Christensen, M. Collins, N. Faull, D. J. Frame, J. A. Kettleborough, S. Knight, A. Martin, J. M. Murphy, C. Piani, D. Sexton, L. A. Smith, R. A. Spicer, A. J. Thorpe, and M. R. Allen. Uncertainty in predictions of the climate response to rising levels of greenhouse gases. *Nature*, 27 January, 2005.

[9] D. A. Stainforth, M. R. Allen, , D. Frame, J. Kettleborough, C. M. Christensen, T. Aina, and M. Collins. Climateprediction.net: A global community for research in climate physics (ch. 12). *Environmental Online Communication (Editor: Arno Scharl)*, 2004.

[10] D. A. Stainforth, J. A. Kettleborough, M. R. Allen, M. Collins, A. Heaps, and J. Murphy. Distributed computing for public-interest climate modelling research. *Computing in Science and Engineering*, 4:82–89, 2002.

[11] D. A. Stainforth, J. A. Kettleborough, A. Martin, A. Simpson, et al. climateprediction.net: Design principles for public-resource modeling research. *Proceedings of the 14th IASTED International Conference, Parallel and Distributed Computing and Systems*, pages 32–38, November 4-6, 2002.

[12] D. A. Stainforth, A. Martin, A. Simpson, C. Christensen, J. Kettleborough, T. Aina, and M. Allen. climateprediction.net: Security principles for public-resource modeling research. *Proceedings of the 13th IEEE International Workshops on Enabling Technologies*, 2004.

[13] K. D. Williams, C. A. Senior, and J. F. B. Mitchell. Transient climate change in the Hadley Centre models: The role of physical processes. *Journal of Climate*, 14 (12):2659–2674, 2001.

Table 1. Typical volunteer computing Application Resource Usage (Pentium IV/1.6GHz laptop)

Application	Memory Use (MB)	Disk Use (MB)	Completion Time (hours)	Total # Users (millions)
climate*prediction*.net	50	600	840	.045
SETI@home	20	3	6	5
folding@home	5	3	12	.5
distributed.net	5	1	8	.25

Table 2. Typical BOINC Application Workunit Times (Intel Xeon/3.4GHz)

Application	Min CPU Time (D:H:M:S)	Avg CPU Time (D:H:M:S)	Max CPU Time (D:H:M:S)	Sample Size
climate*prediction*.net	20:12:06:38	21:06:49:08	21:22:50:45	3
SETI@home	0 (did not start)	00:02:28:48	00:06:40:43	751
predictor@home	00:01:09:32	00:01:09:58	00:04:43:49	1521
einstein@home	00:09:47:17	00:10:37:07	00:16:26:38	189
LHC@home	0 (did not start)	00:01:07:35	00:12:13:03	598

courtesy of Paul Buck, website http://boinc-doc.net/site-misc/average-processing-time.php5.

Table 3. Launch Week Group - User/Machine Attrition Rate

	Original CPDN	CPDN/BOINC
Launch Date	12/9/2003	29/08/2004
New Users Launch Week	27263	8570
1st Trickle Users *(made it to 1.4% completion)*	18373	6038
User Attrition	32.6%	29.5%
New Machines	51877	13301
1st Trickle Machines	19604	8285
Machine Attrition	62.2%	37.7%
Completed Runs	14895	8321
Completion Rate (per all user)	54.6%	97.1%
Completion Rate (per all mach)	28.7%	62.6%
Completion Rate (per 1st T user)	81.1%	137.8%
Completion Rate (per 1st T mach)	76.0%	100.4%
Active After 6 Mths		
Users	2101 (7.7%)	2894 (33.8%)
Machines	2195 (4.2%)	3748 (28.2%)

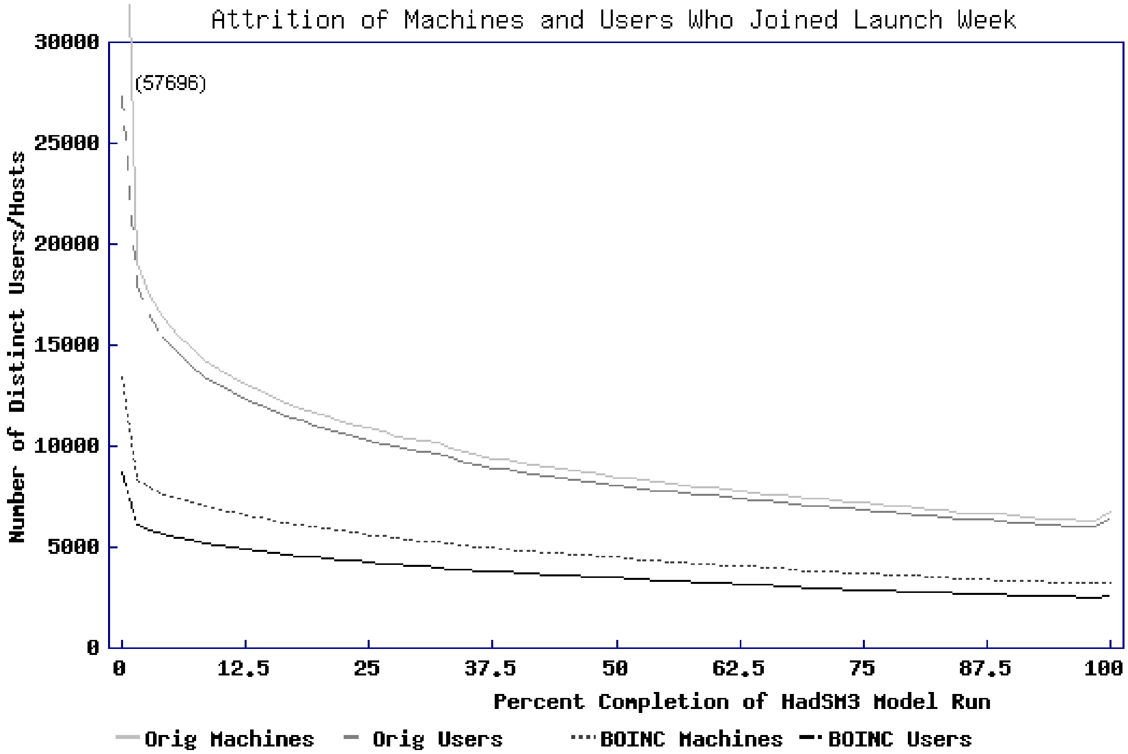

Figure 1. User/Host Attrition by Model Completion

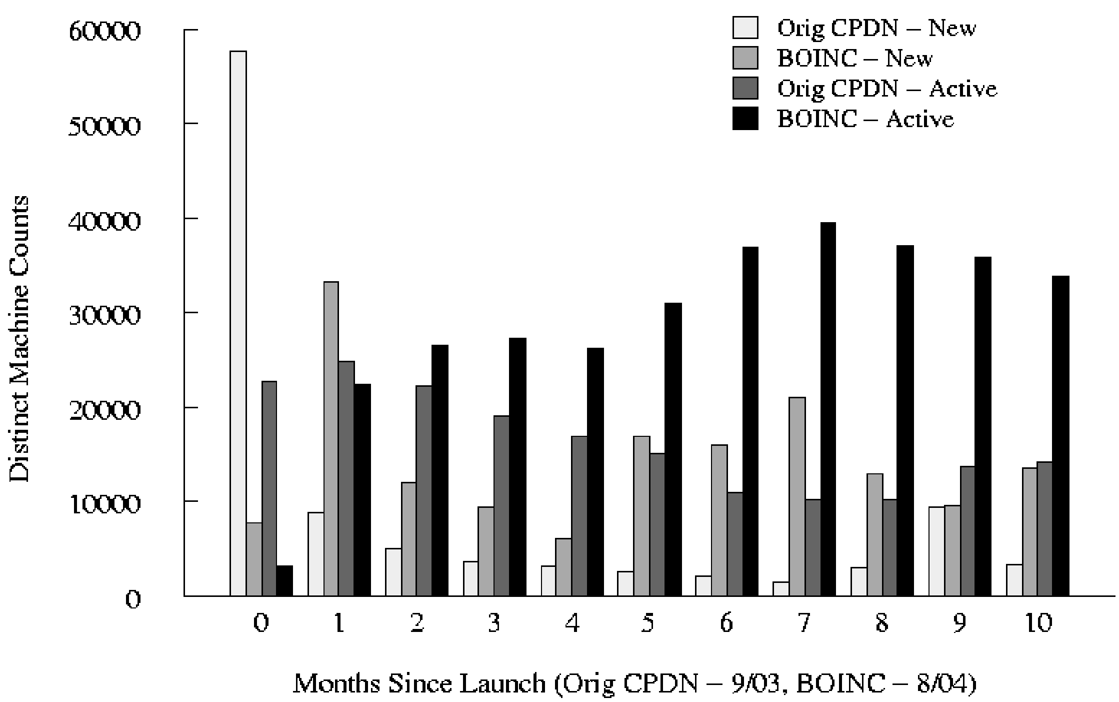

Figure 2. New/Active Machines per Month