# Automated Grid Probe System to Improve End-To-End Grid Reliability for a Science Gateway

Lynn K. Zentner[1], Steven M. Clark[2], Krishna P. C. Madhavan[1,3], Swaroop Shivarajapura[1], Victoria Farnsworth[1,2], Gerhard Klimeck[1,4]
[1]Network for Computational Nanotechnology, Birck Nanotechnology Center,
[2]Rosen Center for Advanced Computing, ITaP,
[3]School of Engineering Education,
[4]School of Electrical and Computer Engineering,
Purdue University, West Lafayette,IN 47907
{lzentner, clarks, cm, swaroop, vfarnsworth, gekco}@purdue.edu

## ABSTRACT
In 2010, the science gateway nanoHUB.org, the world's largest nanotechnology user facility, hosted 9,809 simulation users who performed 372,404 simulation runs. Many of these jobs are compute-intensive runs that benefit from submission to clusters at Purdue, TeraGrid, and Open Science Grid (OSG). Most of the nanoHUB users are not computational experts but end-users who expect complete and uninterrupted service. Within the ecology of grid computing resources, we need to manage the grid submissions of these users transparently with the highest possible degree of user satisfaction. In order to best utilize grid computing resources, we have developed a grid probe protocol to test the job submission system from end to end. Beginning in January 2009, we have collected a total of 1.2 million probe results from job submissions to TeraGrid, OSG, Purdue, and nanoHUB compute clusters. We then utilized these results to intelligently submit jobs to various grid sites using a model for probability of success based in part on probe test history. In this paper we present details of our grid probe model, results from the grid probe runs, and a discussion of data from production runs over the same time period. These results have allowed us to begin assessing our utilization of grid resources while providing our users with satisfactory outcomes.

## Categories and Subject Descriptors
B.8.2: Performance Analysis and Design Aids; C.4 [**Performance of Systems**]: Measurement Techniques; J.2 Engineering

## General Terms
Algorithms, Measurement, Performance, Reliability

## Keywords
Science Gateway, Nanotechnology, Simulation, nanoHUB, Grid Computing, Performance Monitoring, HUBzero

## 1. INTRODUCTION
The science gateway nanoHUB.org hosts over 2600 content items including over 190 simulation tools. In 2010 alone, 9,809 users performed over 370,000 simulation runs [1]. The vast majority of these runs are in the form of rapid, interactive simulations to guide learning, intuition, and experimental research. These simulation runs are characterized by their extremely quick turnaround times. Such runs can execute in our HUBzero-based virtual execution hosts that form the core of the nanoHUB middleware system. Our virtual host system has demonstrated the simultaneous support of over 480 simultaneous users in a rather modest cluster computer. However, some of our tools require significant computational efforts and can strongly benefit from true parallel execution in an MPI environment on hundreds if not thousands of cores, as well as modest parallel runs that can execute well on many serial machines. These simulations need to be dispatched to computational engines external to the central nanoHUB engine. The typical nanoHUB user is not a computational expert who is familiar with grid submission processes and details. Like many science gateway users, our user does not want to become a computational expert, but rather, expects a computational service that delivers transparent and complete results as rapidly as possible. As our online simulation facility became more established, we observed an increased number of requests for more computationally intensive simulations engines. Matching users and their particular simulation requests with the most appropriate and effective computational host is a critical service nanoHUB needs to deliver. Via community accounts, nanoHUB has access to several Network for Computational Nanotechnology (NCN) cluster computing resources, including the Purdue-led DiaGrid, several TeraGrid resources, and OSG.

About four years ago we began to connect nanoHUB to external grid resources via standard protocols (various versions of Globus and CondorG). At that time we observed ongoing user frustration with failed job submissions and long wait times for executions on external compute resources. In general, our users' experiences with external grid resources were running completely counter to the Quality of Service we aim to provide through nanoHUB.org. No true monitoring and failure analysis systems were in place that would give us systematic insights into understanding the failure mechanisms and possible improvements to enhance users' experiences. To effectively utilize the available grid computing resources, testing and analyses were required to determine grid site health (including communication paths, communication software, and the actual status of the compute resource). A joint nanoHUB-OSG Task Force was formed in November 2008 to address some of these issues. Our probe test procedures evolved, in part, based on interactions with the task force. The end goal of the probe test procedures was to allow us to direct user job submission based on these test results.

There are a variety of resources, some specific to TeraGrid, providing system information, wait time and start time data,


*TeraGrid '11*, July 18-21, 2011, Salt Lake City, Utah, USA.

queuing time prediction, as well as detection of problems in the grid infrastructure. Karnak [2], QBETS [3], and TeraGrid's Integrated Information Services (IIS) [4] provide information regarding queues and system performance while INCA [5] provides automated testing and monitoring of the grid infrastructure. Sivagnanam and Yoshimoto [6] evaluated the performance of a variety of resource selection tools against random submission for real jobs sent to TeraGrid resources and found that the tools do provide improvement over random submission. However, the variety of computing resources utilized by nanoHUB required a more general approach, which we developed and describe in this paper. We are engaged with multiple grids and other resource providers and the level of monitoring and reporting varies across these sites. The reporting mechanisms if they exist are heterogeneous. The grid-hosted monitoring programs address the status of their hardware/software and access methods, a necessary but not sufficient condition for determining individual gateway access. Opportunist resource providers often will have different quotas and/or priorities for different submitters. These limits can change dynamically and may not be known to the submitters. In the cases where access to large multi-node jobs are allowed, different users or submitters may have access to different queues - again with different limits and priorities. In addition, a native probe system provides information not only on the health of the grid, but information regarding problems within our own system and job submissions. Thus, probe results obtained through our own testing may be more indicative of the actual behavior we will see in our submissions than predictions by grid-side systems.

In January 2009, we began sending grid probe tests to all computing resources utilized by nanoHUB, both within and outside Purdue University. To date, we have collected 1.2 million probe results. Based on these probe results, we have begun to direct our job submission and adjust our use of various grid computing resources to maximize quality of service and also enhance users' experiences when interacting with nanoHUB.org.

## 2. THE GRID PROBE SYSTEM

The grid probe system was developed to test the job submission system end-to-end—from the nanoHUB tool environment to the remote site and back. There are several steps in the job submission chain and any broken link can lead to a job failure. A successful probe is a job that is submitted to a specific site and returns without error. The faster the turnaround time for a successful job, the better the result. Given the number of potential sites available for nanoHUB job submissions, it was clear that the probe process needed to be automated. Two daemon processes described below manage the process of probe submission and the subsequent result collection.

The first daemon, *probeLauncher*, is responsible for submitting probes to all sites that could serve as job execution hosts for nanoHUB jobs. There are four major collections of these execution hosts: TeraGrid, OSG, Purdue University campus clusters, and nanoHUB-operated clusters. The combined resources of these collections provide a heterogeneous set of computational platforms for the execution of several scientific applications. The purpose of the probe is twofold: (1) to determine if a site is currently accessible and operational, and (2) to measure the speed at which a given site can return computational results. The probe itself is a simple shell script requiring a single core that is submitted through the nanoHUB job submission process in the same manner as a production application run. In most cases the probe will wait in a batch queue pending execution on the remote site, execute in a short time, and return results. Any waiting time in a batch queue increases the turnaround time of the job, and the actual execution time is designed to be negligible. The launch daemon schedules the next probe submission upon completion of any probe job. The time between probe job completion and a subsequent submission to the same site is an externally configurable parameter set to 30 minutes. There is a tradeoff to be considered between overwhelming the system with probe runs and the accuracy needed to determine the likelihood of a successful production run completion.

The second daemon, *probeMonitor*, is responsible for collecting the results of probe jobs and providing the results on-demand to production application runs. The raw data input from the *probeLauncher* is processed to produce a score for each site. The raw data consists of exit status (pass/fail), turnaround time, and completion time. The turnaround time is discretized on a non-uniform scale with the following endpoints: a turnaround time of less than five minutes rates a score of 100, while a turnaround time greater than six hours rates a score of 0. Clearly our scale is relatively arbitrary; we chose it to reflect our users' needs to get quick turnaround through a balanced system of available computing resources. The relative scores of the sites are used as a partial basis for site selection on a job-by-job basis. In addition to the site score, the age of the score is also considered. If a score is old, it indicates that the subsequent probe has not yet returned, implying that the current response time of a site is slow. Thus, the score of the next probe is likely to be low and a site that responds faster should be used. The probe score contributes to the Condor rank calculation when submitting jobs to grid resources. Condor applies additional measures to avoid flooding a site with jobs during any site matchmaking cycle. As an additional measure the rank of any site can be modified through the site's classAd. This final measure allows for administrator intervention to reflect factors not measured or considered by the probe process. A use case for this feature is to account for a site's preemption policy. If a site frequently allows jobs to start but subsequently preempts them, it is wise to downgrade the score for that site.

The results of the probe process are made available through a webpage [7]. The webpage groups results by execution host collection, time frame, and individual site. A breakdown is provided detailing the success rate of each site in summary and timeline form. The results are aggregated every two hours and reported in two formats to highlight both the success rate and turnaround characteristics of the probe runs. The aggregated results are reported for each calendar month as well as the previous 24 hours, 7 days, and 31 days to provide historical perspective.

Figure 1 shows the first report, highlighting the status of each probe run. Each probe run is represented by a point on the strip chart indicating the three possible results.

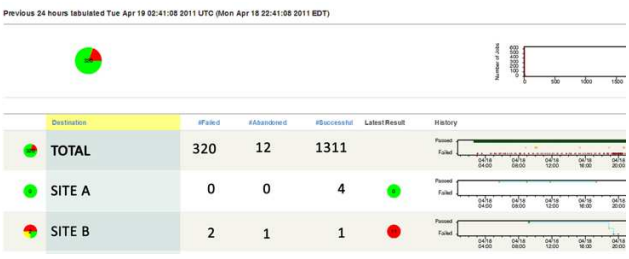


**Figure 1. Status of Probe Runs (Web Image)**

The possible results are:

- Successful (green) – the probe returned with no errors
- Abandoned (yellow) – the probe was either preempted or the maximum wait time was exceeded
- Failed (red) – the probed exited with a non-zero exit status

A quick glance at the pie charts on the left hand side of Figure 1 gives a good indication of those sites that are not working well for probe and, by extension, job submission. The webpage provides a common point of reference when working with site administrators to debug any issues. A quick glance at the strip charts on the right side of Figure 1 can indicate when a problem is occurring simultaneously across many sites. This often indicates that a problem within the nanoHUB infrastructure needs to be addressed.

A separate but complementary table gives a summary of the occurrence of specific exit codes for failed probes across all sites in the collection. The error codes indicate common problems that can occur between nanoHUB and the remote sites as well as those directly related to problems in the nanoHUB infrastructure. Using the error code information, it is possible to more quickly address any problems in the job submission process at a specific site. It is also possible to view the textual output of each failed probe run.

Another report providing further information regarding turnaround times for each site can be obtained by clicking on the time bar at the top of the status report shown in Figure 1.

The time metrics are computed based only on the successful probe runs. The histogram at the top right of each report, shown in part in the upper right of Figure 1, reflects the distribution of turnaround times with a bucket width of five minutes. The histogram is available for the entire collection and for each individual site in the collection.

## 3. ANALYSIS

The large volume of data produced by the grid probe tests can be examined several ways to gain insights into the job submission process. In this section, we present the complete set of results from probe tests to TeraGrid, OSG, and local clusters from the beginning of testing until the present time. We also provide more detailed data highlighting specific time periods, as well as results from production runs.

### 3.1 Probe Results

Figure 2 presents the percent of successful, abandoned, and failed probe tests over all TeraGrid sites for the period spanning January 2009 through April 2011, as well as the average response time for each month. Several months have greater than 90% successful runs, including February 2009, August 2009, and the period from July through October 2010. The highest failure rate occurred in February 2010, with nearly half the probe tests resulting in failure.

Figure 3 presents similar results for OSG. OSG probes showed about a 90% success rate in July 2009 and failure rates of close to 40% in January and November 2009, as well as February and November 2010 and January 2011.

Figure 4 presents results for the execution hosts in the local cluster. Probe failure from the local cluster is noticeably lower than that for the grid sites, with failure rates of under 10% (except for the months of July, August, and November 2010, which had failure rates of 15–20%).

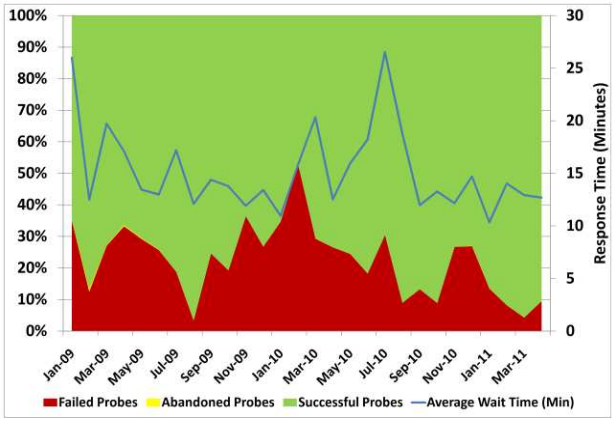


**Figure 2. TeraGrid Probe Results**

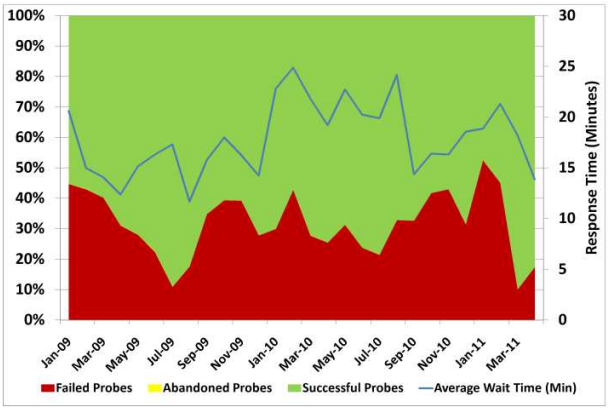


**Figure 3. Open Science Grid Probe Results**

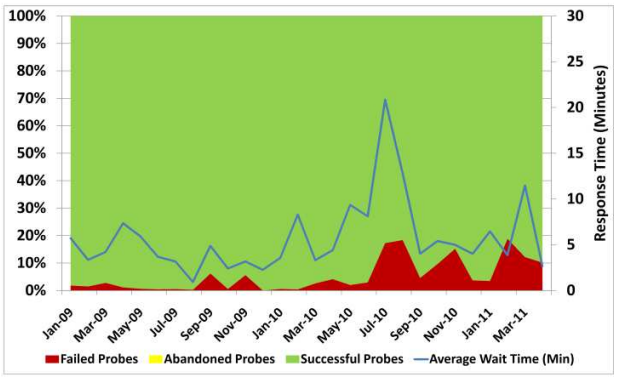


**Figure 4. Local Cluster Probe Results**

In each of the figures in this section, the average response time in minutes for each month is also plotted as a blue line. Interestingly, there does not seem to be a clear correlation between average response time and the health of the site. In fact, at times when the site returns fewer failed submissions, the response time can be high compared to the response time when the site is returning more failed submissions. One possible explanation is that when a site is fairly healthy, that status may result in more job submissions to that site, increasing wait time.

## 3.2 Detailed Data Overview

Figure 2 shows that in February 2009, probes to TeraGrid were successful about 90% of the time and the average response time for the probes was about 15 minutes. Figure 5 shows a detailed view of the results for February 2009. The gap in results on February 8, 2009 indicates a day when no probes were initiated. This corresponds to an outage on nanoHUB.org for scheduled maintenance. Though the average success rate for this month was about 90%, there are instances during the month where it decreases to as low as 70%. Response times up to 20 minutes were logged, frequently on days with high probe success rates.

Figure 2 shows that in February 2010, approximately 50% of probe tests failed. Figure 6 shows a detailed view of that month. For several days, all probes sent to TeraGrid failed. This was not due to a problem with TeraGrid, but to a power outage at Purdue that affected the machine that was used as the Certificate Revocation List (CRL) server for our X509 certificates. Similar failures were returned from probes sent to the OSG during this time period. The rest of the month showed failure rates varying from 30% to just under 60%.

Clearly these results call for further optimization of our monitoring system, so that failures can be found, identified, and fixed more rapidly. These results suggest a general need for infrastructure monitoring and support, potentially including dedicated gateway personnel.

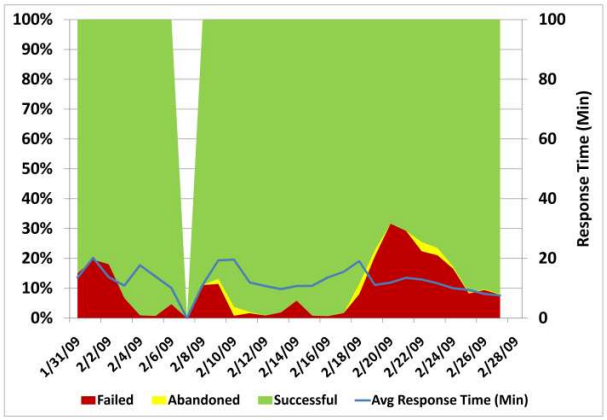


**Figure 5. TeraGrid Probe Results, Feb 2009**

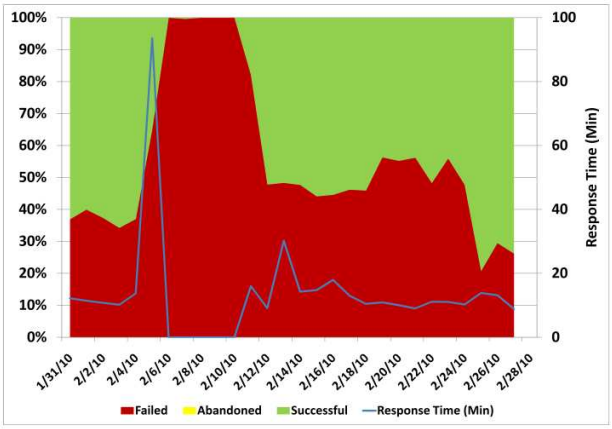


**Figure 6. TeraGrid Probe Results, Feb 2010**

## 3.3 Probe Scores

As discussed in Section 2, the grid probe results are assigned a score from 0 to 100 based on a non-uniform scaling of turnaround time. This probe score functions as one part of the Condor rank calculation, in conjunction with human intervention ability and probe score age. Job submission is also based on job failure history on a per job basis. That is, a job that has failed at a given site will not be resent to that site.

The figures that follow present a more detailed spread of scores for each resource collection in a given month.

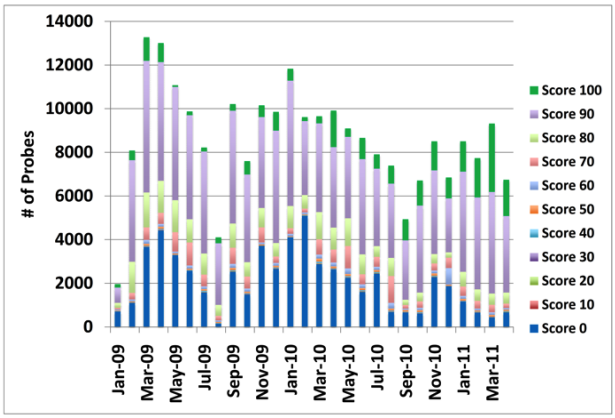


**Figure 7. TeraGrid Probe Scores**

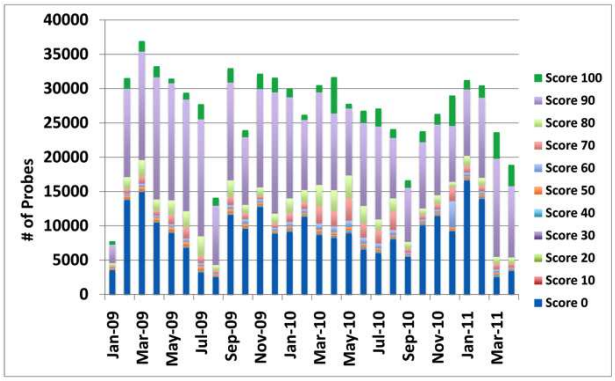


**Figure 8. Open Science Grid Probe Scores**

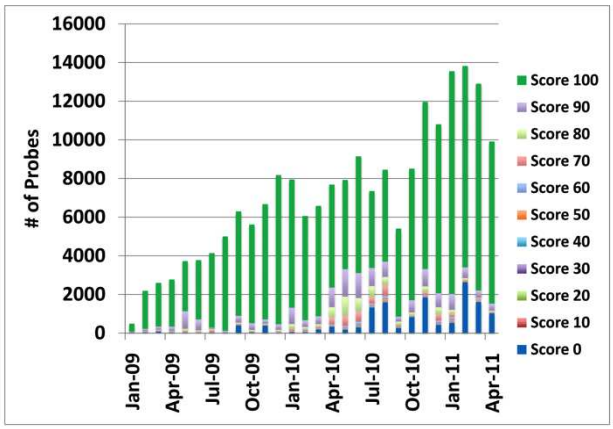


**Figure 9. Local Cluster Probe Scores**

A large proportion of the scores for the local cluster are high, resulting in a large number of jobs directed to those sites. Both OSG and TeraGrid demonstrate a significantly smaller proportion of scores at 100 as well as a larger proportion of scores at 0. The impact of these scores is shown in the number of production runs sent to each collection, summarized in Table 1.

## 3.4 Production Run Results

Figure 10, Figure 11, and Figure 12 present the results from actual production runs on a monthly basis from January 2009 through April 2011 for all collections utilized by nanoHUB.org. The graphs provide the proportion of runs that failed due to grid errors, failed for other reasons (including being abandoned), and successfully completed runs. The right axis on each plot shows the number of runs sent to each resource in a given month.

It can be seen from Figure 10 that TeraGrid experienced high failure rates due to grid errors over many months in 2010 and also that the overall number of jobs sent to TeraGrid in a given month were significantly lower than the number of jobs sent to OSG and the local cluster. Figure 11 shows a smaller percentage of jobs failing due to grid errors, with the exception of February 2010, but with a consistently higher percent of jobs failing due to other errors. The local cluster, shown in Figure 12, experienced the highest number of job submissions with relatively low grid failure rates and more moderate rates of other failures than OSG.

Patterns can be seen in the data in all three plots indicating that as grid errors increase, jobs submitted to that resource decrease, indicating that the probe tests are resulting in production jobs being directed to resources with more likelihood of a successful run. However, jobs still fail due to grid errors, despite testing. The production jobs require file transfer, and in spite of testing, ever transfer is subject to failure. There are also differences between the probe and production job characteristics, such as longer run times and multiple cores, which can result in wait time exceeded errors. Production code requires modules to be loaded to define location of various libraries, for instance, and custom configuration on an application basis is not possible through GRAM job submission. Probe jobs require no such modules. We are considering submitting more sophisticated probes for sites with such configuration limitations, requiring the option of custom probes for different sites, which has recently been made possible.

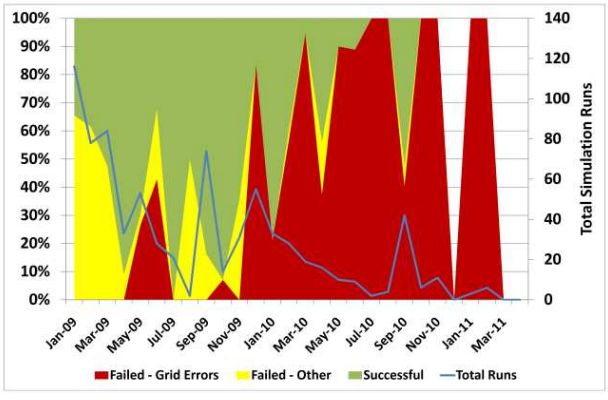


**Figure 10. TeraGrid Production Run Results**

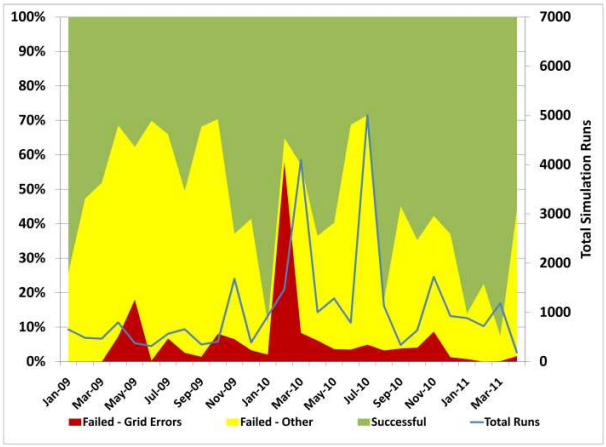


**Figure 11. Open Science Grid Production Run Results**

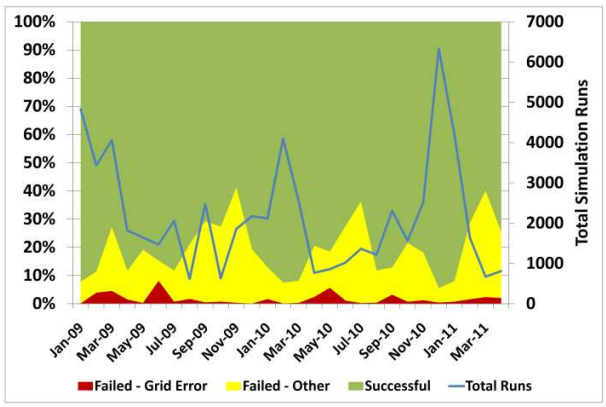


**Figure 12. Local Cluster Production Run Results**

Table 1 provides the total number of production runs sent to each resource collection over the period used for this study. The number of jobs sent to TeraGrid is quite low compared to OSG and the local cluster. Despite use of the grid probe results, nearly a quarter of the jobs sent to TeraGrid failed due to grid errors, while that rate was 7% and 2% for OSG and the local cluster, respectively. The overall end-to-end success rate takes into account user input which could crash the science code. We observe that both TeraGrid and OSG had similar success rates of 49% and 52%, respectively, while submissions to the local cluster succeeded at a rate of 84%.

**Table 1. Totals for Simulations on Each Collection**

| | TeraGrid | | OSG | | Local Cluster | |
|---|---|---|---|---|---|---|
| Total Runs | 778 | | 29,448 | | 61,124 | |
| Successful Runs | 385 | 49% | 15,276 | 52% | 51,259 | 84% |
| Total Failed | 393 | 51% | 14,172 | 48% | 9,865 | 16% |
| Grid Errors | 186 | 24% | 2,193 | 7% | 922 | 2% |

## 4. CONCLUSIONS

The results presented in this paper provide a preliminary overview of our experience with the broad set of computing resources available to nanoHUB.org users for production simulation runs. Our objective is to match our users with computing resources appropriate to their needs that will return results successfully in a timely fashion.

The grid resource testing done for the purposes of improving nanoHUB.org job submission has also benefited the larger grid community. On multiple occasions, problems revealed by the probe testing affected other gateways or communities in addition to nanoHUB.org. The probe results have been used as a communication medium between nanoHUB.org and all pertinent (cluster/grid) service organizations to help resolve problems.

The extensive testing we have done shows that grid submission still involves a complex process that can fail over time and requires detailed monitoring and care. A failover and ranking system as developed here can help to overcome some of the detectable infrastructure failings by preventing submission to sites that are not returning timely, successful results at any given time. However, dedicated staff time is needed to support the production end-to-end service on such networked infrastructures, as some of the catastrophic failures are systemic and require human intervention. Even with our best effort and engagement by OSG virtual organization (VO) support in particular, we were not able to eliminate these sporadic but catastrophic errors.

Our initial evaluation of these results indicates that the existing policy to maximize use of our local cluster is a good choice at this time. However, we anticipate an increase in users requiring large numbers of cores as they scale up jobs that, for instance, utilize compute intensive programs such as MIT Electromagnetic Equation Propagation (MEEP), abinit, and Large-Scale Atomic/Molecular Massively Parallel Simulator (LAMMPS). As the core requirements for these runs expand, it will become increasingly important for us to direct future jobs to the grid resources that can most appropriately handle them.

We will continue to improve our probe system to enable reliable end-to-end submission into the grid, which already includes preemptive submission to multiple compute sites combined with a job cancellation upon receipt of the first completed result. There is no off the shelf monitoring process for a gateway, given that the gateway submission infrastructure is a customized process. We believe that reliable submission into "the cloud" will require similar monitoring and evaluation systems, and efforts to tie nanoHUB into this infrastructure are under way.

## 5. ACKNOWLEDGMENTS

Mark S. Lundstrom founded nanoHUB.org in 1998. In 2005, Michael McLennan created the Rappture Toolkit and Rick Kennell wrote the scalable middleware of HUBzero that, respectively, enable and power interactive nanoHUB simulations. The Network for Computational Nanotechnology (NCN) manages nanoHUB.org and is funded by NSF Award # EEC-0228390. We also gratefully acknowledge collaborations and support from the OSG VO team and the TeraGrid Science Gateway group.

## 6. REFERENCES

[1] nanoHUB usage website: http://nanohub.org/usage/overview/year

[2] Karnak Prediction Service: http://karnak.teragrid.org/karnak/index.html

[3] Daniel Nurmi, John Brevik and Rich Wolski. 2008. QBETS: Queue Bounds Estimation from Time Series. *Lecture Notes in Computer Science: Job Scheduling Strategies for Parallel Processing.* Springer. Volume 4942 (2008), 76-101. DOI= http://dx.doi.org/10.1007/978-3-540-78699-3_5.

[4] Lee Liming, John-Paul Navarro, Eric Blau, Jason Brechin, Charlie Catlett, Maytal Dahan, Diana Diehl, Rion Dooley, Michael Dwyer, Kate Ericson, Ian Foster, Ed Hanna, David L. Hart, Chris Jordan, Rob Light, Stuart Martin, John McGee, Laura Pearlman, Jason Reilly, Tom Scavo, Michael Shapiro, Shava Smallen, Warren Smith, and Nancy Wilkins-Diehr. 2009. TeraGrid's integrated information service. In *Proceedings of the 5th Grid Computing Environments Workshop* (GCE '09). ACM, New York, NY, USA, Article 8, 10 pages. DOI=http://doi.acm.org/10.1145/1658260.1658271.

[5] Shava Smallen, Catherine Olschanowsky, Kate Ericson, Pete Beckman, and Jennifer M. Schopf. 2004. The Inca Test Harness and Reporting Framework. In *Proceedings of the 2004 ACM/IEEE conference on Supercomputing* (SC '04). IEEE Computer Society, Washington, DC, USA, 55-. DOI=http://dx.doi.org/10.1109/SC.2004.56.

[6] Sivagnanam, S. and Yoshimoto, K. 2010. TeraGrid resource selection tools: a road test. In *Proceedings of the 2010 TeraGrid Conference* (August 2-5, 2010, Pittsburgh, PA). TG '10. DOI= http://doi.acm.org/10.1145/1838574.1838594.

[7] nanoHUB grid probe results website: http://nanohub.org/usage/gridprobe