

# How to Run a Million Jobs in Six Months on the NSF TeraGrid

Edward Walker, David J. Earl, and Michael W. Deem

**Abstract**— In June 2006 a team of researchers began submitting workflows across the distributed multi-user clusters on the NSF TeraGrid. The goal of their scientific study was to discover as many new hypothetical zeolite crystalline structures as possible. In just over six months, the researchers succeeded in completing over a quarter of a million workflows, comprising over a million jobs. In all, the team consumed a million computational hours, harnessing over 200 TFLOPS of distributed resources, to help populate a database of over three million new crystalline structures. This paper describes the challenges faced by the researchers and the submission system implemented by them to overcome these challenges.

**Index Terms**— Workflows, distributed computing, parameter sweep, material science

---

## 1 INTRODUCTION

This paper documents the experience of a group of computer and material science researchers who are populating a public scientific database of hypothetical zeolite crystalline structures [1]. These zeolite crystalline structures are a class of porous material widely used as catalysts and ion exchangers in many important applications and processes in science and industry. The goal of the material science researchers was to run computer simulations to identify zeolite structures that potentially have real counterparts in nature. These structures could then be added to a in a public database of zeolites, enabling the collective community of material scientists to synthesize and experiment with novel materials, applications and processes.

To achieve their scientific goal, the research team was awarded over two million computational hours on the NSF TeraGrid to perform this important scientific study. Over the course of the allocation period, one million computational hours were designated for the cycle-scavenging Condor pool at Purdue University, and one million to all other HPC (High Performance Computing) resources on the TeraGrid [13].

This paper describes how the latter allocation was efficiently used by the team of researchers. In particular, this paper describes how the team aggregated the distributed resource at NCSA (National Center for Supercomputing Applications), SDSC (San Diego Supercomputing Center), ANL (Argonne National Laboratory) and TACC (Texas Advanced Computing Center) into personal computing laboratories for performing the calculations required by their scientific study. The HPC clusters used in contributing resources to these personal computing laboratories were shared multi-user systems, supporting hundreds of other researchers across a diverse range of scientific disci-

plines, running jobs of multiple modalities with different job size and run time requirements.

The researchers were able to create their personal computing laboratories using the middleware tool called MyCluster [2]-[3]. The tool provided a number of significant advantages to the submission system which was developed to manage the simulation jobs in their scientific study. First, the submission system was able to submit and manage jobs across eight different HPC clusters from a central client workstation, through a single interface, over a continuous period of over six months. Second, the submission system was able to maximize their job execution throughput using MyCluster's ability to adaptively aggregate resources based on local and global load conditions. Third, the system allowed transient network outages, and periodic site reboots, to be tolerated during long running periods of their calculation runs. Fourth, the submission system allowed the use of workflow tools to automate, orchestrate and throttle the simulation jobs for the manageability and efficiency of execution of the large number of jobs in the scientific study.

The rest of this paper will be organized as follows. Section 2 describes in more detail the motivation of the scientific study conducted by the material science researchers. Section 3 discusses the challenges in conducting their scientific study on existing HPC cyberinfrastructures. Section 4 outlines the solution developed by the team of researchers to address these challenges. Finally, section 5 concludes this paper.

## 2 ZEOLITE CRYSTALS

Zeolites are crystalline microporous materials that have found a wide range of uses in industrial applications. They are used as catalysts, molecular sieves, and ion-exchangers, and are expected to be of importance in a wide range of nanoscale applications. A typical example is ZSM-5, used as a cracking co-catalyst in the refinement of crude oil.

Classical zeolites are aluminosilicates. The basic building block is a  $TO_4$  tetrahedron. Usually  $T = Si$ , although substitution of the silicon with aluminum, phosphorus, or other

- 
- Edward Walker is a Research Associate at the Texas Advanced Computing Center, The University of Texas at Austin. Email: ewalker@tacc.utexas.edu
  - David J. Earl is an Assistant Professor at the Department of Chemistry, University of Pittsburgh. Email: dearl@pitt.edu
  - Michael W. Deem is the John W. Cox Professor of Biochemical and Genetic Engineering and Professor of Physics and Astronomy, Rice University. Email: mdeem@rice.edu.

metals is common. The tetrahedral species is commonly denoted by T when one is concerned with structural, rather than chemical, properties of the zeolite. From these simple tetrahedral building blocks, a wide range of porous topologies can be constructed.

Roughly 180 framework structures have been reported to date [5]. There is therefore a tremendous demand for new zeolite structures with novel properties. The main goal of the hypothetical zeolite research project is therefore to generate topologies that are chemically feasible and are predicted to be of industrial importance to enable material scientists to design and target materials with these new properties.

### 3 COMPUTATIONAL CHALLENGE

The material science researchers used simulation methods to explore the space of possible zeolite structures. For each zeolite crystallographic space group (230 in total), the range of possible unit cell sizes ( $a, b, c, \alpha, \beta, \gamma$ ) were explored. Discrete jumps in the lengths and angles defining a unit cell were also used to perform a thorough search for potential zeolite structures. Also, for each unit cell size the density of tetrahedral atoms and the number of crystallographically unique tetrahedral atoms were varied to further extend the search through crystallographic space for feasible structures.

*Challenge 1: Millions of computer simulations need to be run*

To optimize a zeolite figure of merit for each parameter set proposed in the study, the researchers used a biased Monte Carlo simulated annealing method. The researchers needed to run their calculation for all possible crystal configurations. The goal was therefore to submit millions of simulations to cover this large parameter space. Submitting millions of simulations to a job scheduler queue puts a considerable burden on any system and runs against many site-specific policies. Also, monitoring these jobs and checking their output results takes a considerable amount of effort by the researchers. Therefore, some means of throttling the simulation job submissions, taking remedial actions when faults occur, and checking output results when jobs complete were considered critical requirements.

*Challenge 2: Each simulation job was serial in nature*

The material science researchers developed their simulated annealing algorithm to solve their multi-variable, non-linear system. The simulated annealing algorithm allows a series of random "near-by" solutions to be generated over a period of decreasing simulated "temperature", with solutions of increasing optimality generated over time. This annealing process is therefore intrinsically serial in nature because the next step depends on the result of the previous step. Hence, although collectively the entire job ensemble was embarrassingly parallel, each job in the simulation was serial in nature.

The serial nature of each job posed a challenge to the researchers. First, many sites have schedulers configured to favor parallel jobs over serial ones [11]. Furthermore, some

scheduling policies increase the priority of jobs based on their parallel job size, i.e. larger parallel jobs are favored over smaller ones. The rationale behind these scheduling policies is to favor jobs engaged in large computations and ensure that these are not unduly penalized by other smaller jobs in the job queue. However, the hypothetical zeolite researchers were constrained by the intrinsic serial nature of their optimization technique. So a solution was needed to enable their jobs to be equally favored by the local job schedulers at each site.

Second, some sites have a limit to the number of jobs a user can concurrently submit to the local queue. For example, at TACC this job submission limit is 15 jobs per user. Therefore, enabling as many simulations to run per job submission was critical to ensure the scientific study could complete in a timely manner.

*Challenge 3: Simulation jobs had a wide variability in the execution times.*

Each simulation job was composed of a series of simulated annealing computations over a set of 100 seeds. Because of the pseudo-random nature of the algorithm, the expected run time for each job could take anywhere from a few minutes to 10 hours. In some cases, the simulation would never terminate. A majority of the jobs however were expected to complete within 3 to 4 hours.

This huge variability in the run times of the simulation jobs caused some significant problems. First, each simulation job would need to request a CPU resource for 10 hours to ensure the worst case possible run-time requirement could be met. However, if a job ran for only a few minutes, the remaining time left on the CPU would be unsuitable for other simulation jobs with the same run-time requirement. It was therefore critical to re-factor the computation into smaller sub-jobs to reduce the variability of the expected job run-times.

Second, some simulation jobs would never terminate. The researcher therefore had to implement additional processing to detect these types of jobs for early termination. The additional processing involved closely monitoring the run-time of the simulated annealing computation for each seed, and terminating the job if the seed run-times indicated that the job would exceed its 10 hour run-time limit. A side effect of this additional processing was to further prevent running each seed simulation in parallel, because run-times of earlier seeds need to qualify if later seeds can run.

## 4 THE SOLUTION

### 4.1 MyCluster Overview

The solution developed by the researchers to address their computational challenge was based on the MyCluster tool. MyCluster is a production software services on the NSF TeraGrid [6]. The system provides the capability of provisioning over 200 TFlops of distributed resources across the TeraGrid into personal clusters created on-demand. Personal clusters are useful because they can be treated as containers for experimental runs. Faults in an experiment in a

personal cluster container are isolated from other concurrently running experiments. These personal clusters also tolerate faults in the underlying physical infrastructure. MyCluster is able to do so because it deploys semi-autonomous agents to provision resources from each contributing HPC cluster. These agents are able to recover autonomously after periodic site reboots, and survive transient network outages across the wide-area network, allowing the personal clusters to remain unperturbed while experiments are running. Finally, MyCluster allows users to submit and manage jobs across the TeraGrid from a central point, using a single well-known job management interface. The current production version of MyCluster on the NSF TeraGrid allows users to use the Condor job management interface [7]-[8] for submitting and managing jobs.

#### 4.2 Creating personal clusters with job proxies

MyCluster allowed the researchers to create a personal Condor cluster from a workstation at TACC. The workstation served as the submission point for all jobs submitted across the multiple systems on the TeraGrid.

The researchers configured MyCluster to submit parallel job proxies, with job sizes ranging from 20 to 64 CPUs, to the HPC clusters at NCSA, SDSC, ANL and TACC. These job proxies, submitted with a run-time requirement of 24 hours each, then contributed CPUs to the personal cluster when they were run by the local scheduler at a HPC cluster.

The CPUs from the job proxies became part of the personal Condor cluster until the job proxies contributing to them were terminated by the HPC cluster scheduler. To allow the Condor scheduler in the personal cluster to correctly assign jobs to CPUs with sufficient time left to complete them, the job proxies advertised their remaining time left in the HPC cluster in a *TimeToLive* classad [12]. Jobs could then request for CPUs in the personal cluster with a *TimeToLive* requirement that was at least the expected run-time of the job.

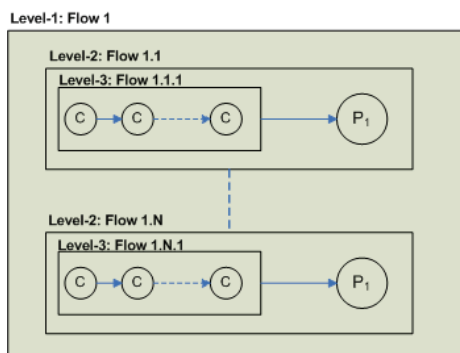


Figure 1. Example multi-level nested workflow

#### 4.3 Multi-level nested workflows

The researchers created nested workflows for executing their simulation experiments. These nested workflows orchestrated and throttled jobs through the personal Condor cluster. A visual representation of this nested workflow is

shown in Figure 1. The level-1 workflow was composed of an ensemble of job chains, each chain representing a pair of jobs in a level-2 workflow. The level-2 workflow was composed of a job for scheduling the simulation, and another for post-processing the output of the simulation. The node responsible for scheduling the simulation spawned a level-3 workflow composed of a chain of five jobs. These five jobs collectively represented the simulated annealing computation for 100 seeds.

Each job in the level-3 workflow computed over 20 seed values, for an expected run time of 2 hours per job. Therefore each job requested a CPU resource in the personal cluster with at least 2 hours in its *TimeToLive* value. Multiple level-3 workflow jobs were therefore able to run on each CPU acquired by the job proxies.

Each level-3 workflow job also measured the run-time of the simulated annealing computation for each seed, to ensure early detection of simulations not expected to terminate. If four consecutive seeds resulted in computation times exceeding the expected run-time per seed (six minutes), the job was terminated, and a special exit code returned to the level-2 workflow. The level-2 workflow then terminated and an error message was logged. However, if the level-3 workflow completed successfully, the post-processing job of the level-2 workflow was triggered. This post-processing job was then scheduled locally on the client workstation to check the validity of the computed results. If an error was detected in the computed results, an error message was logged; otherwise, the result files were archived. The level-2 workflow then exited.

#### 4.4 Running workflows in personal clusters

The team of researchers grouped each experimental run into space groups. Each space group described a collection of potential crystalline structures within a range of possible cell sizes, and was typically composed of between 6000 to 30000 members. A command line tool was created to parse through each space group to generate the level-1, level-2 and level-3 workflow definition files in the Condor DAGMan tool format [14]. A personal Condor cluster was then created from the TACC workstation, and the level-1 workflows submitted to the DAGMan tool running in this personal cluster.

The DAGMan tool was configured to only submit from 350 to 500 jobs to the queue in the personal Condor cluster. This ensured that the client workstation was never overwhelmed by having the personal Condor cluster unnecessarily schedule thousands of jobs simultaneously.

Screen snapshots of a personal Condor cluster created from the workstation at TACC are shown in Figure 2 and **Error! Reference source not found.** MyCluster was used to provision CPUs from HPC clusters at NCSA, SDSC, ANL and TACC. Specifically, the systems used are listed in Table 1.

Table 1. Multi-user TeraGrid systems used for provisioning resources into personal clusters

TeraGrid Site	HPC System	Architecture
NCSA	tungsten	IA-32
	mercury	IA-64
	cobalt	IA-64
SDSC	tg-login	IA-64
ANL	tg-login	IA-64
	tg-login-viz	IA-32
TACC	lonestar	X86_64

The personal Condor cluster, shown in the snapshots, was created to process space group 15\_1, which was composed of 30,000 members. This particular experiment was conducted for a period of over a week from 11-Oct-2006 to 23-Oct-2006.

Figure 2 shows the expanding and shrinking Condor cluster over time, acquiring IA32, IA64 and X86\_64 CPU resources for the workflow jobs.

Space Group 15\_1 Condor Pool Machine Statistics for Week

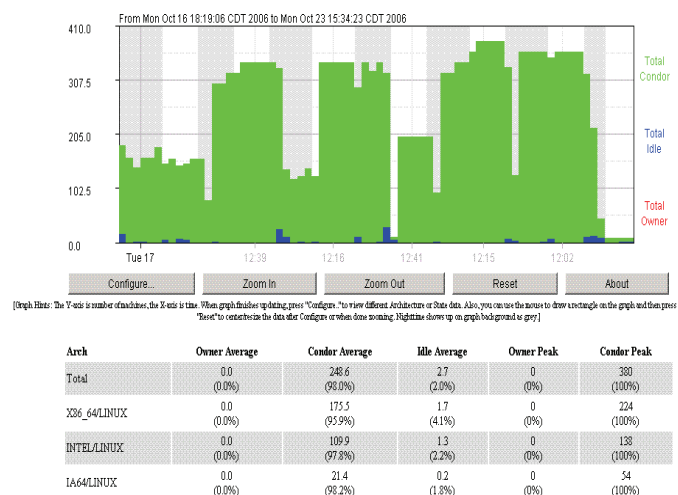


Figure 2. Expanding and shrinking personal condor cluster aggregating resources from NCSA, SDSC, ANL and TACC over a two week period.

**Error! Reference source not found.** shows the jobs in the Condor queue. The total number of jobs in the queue never exceeded 380 jobs. This was because the DAGMan tool was configured to throttle the submission of jobs to the Condor scheduler as previously described. This prevented the over-consumption of the local workstation CPU resource.

Finally, Figure 4 shows a hypothetical zeolite crystalline structure that was discovered by the computational study. The study has discovered over three million such structures to date.

Space Group 15\_1 Condor Pool User Statistics for Week

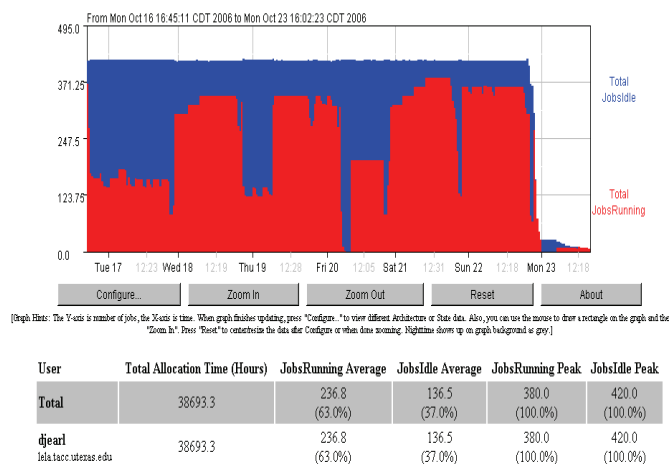


Figure 3. Running and pending jobs in personal condor cluster

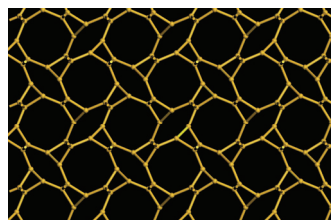


Figure 4. Newly discovered zeolite crystalline structure.

## 5 CONCLUSIONS

The scientific study has added over three million hypothetical zeolite crystalline structures to a publicly-accessible database. One contributing factor in this achievement was the capability of the submission system developed by the researchers to consume over one million computational hours across the multi-user HPC systems on the TeraGrid. Simulation jobs submitted through the Purdue cycle-scavenging Condor pool, and to local clusters at the researcher's home institutions, also contributed to this achievement. However, the ability to harness the large computation capabilities on the NSF TeraGrid accelerated their search in parallel with these other efforts.

The MyCluster system was also central in enabling the success of this study. The system is currently widely deployed on the TeraGrid and future enhancements include allowing users to select the Sun Grid Engine (SGE) [10] and OpenPBS [9] job management interfaces for interacting with jobs submitted across the heterogeneous clusters on the TeraGrid. Furthermore, a wide-area distributed filesystem XUFS [4] will be integrated into MyCluster, allowing jobs transparent access to files in the submission directory from across remote sites. These enhancements will ensure more productivity for researchers using the tool in the future.

## REFERENCES

- [1] D. J. Earl, and M. W. Deem, "Toward a Database of Hypothetical Zeolite Structures", Eduardo Glandt special issue, *Industrial and Eng. Chem. Research*, 54, 2006, pp. 5449–5454.
- [2] E. Walker, J. P. Gardner, V. Litvin, and E. L. Turner, "Personal Adaptive Clusters as Containers for Scientific Jobs", accepted for publication in *Cluster Computing*, Springer.
- [3] E. Walker, J. P. Gardner, V. Litvin, and E. L. Turner, "Creating Adaptive Clusters in User-Space for Managing Scientific Jobs in a Widely Distributed Environment", in *Proc. of IEEE Workshop on Challenges of Large Applications in Distributed Environments (CLADE'2006)*, Paris, July 2006.
- [4] E. Walker, "A Distributed File System for a Wide-Area High Performance Computing Infrastructure", in *Proc. of the 3<sup>rd</sup> USENIX Workshop on Real, Large Distributed Systems (WORLDS'06)*, Seattle, Nov 2006.
- [5] International Zeolite Associate, <http://www.iza-online.org>
- [6] MyCluster TeraGrid User Guid, <http://www.teragrid.org/userinfo/jobs/gridshell.php>
- [7] Condor, High Throughput Computing Environment, <http://www.cs.wisc.edu/Condor/>
- [8] M. Litzkow, M. Livny, and M. Matka. Condor – A Hunter of Idle Workstations, In *Proc. of the International Conference of Distributed Computing Systems*, pp. 104–111, June 1988.
- [9] Portable Batch System, <http://www.openpbs.org>
- [10] Sun Grid Engine, <http://gridengine.sunsource.net/>
- [11] TeraGrid site scheduling policies, [http://www.teragrid.org/userinfo/guide\\_tgpolicy.html](http://www.teragrid.org/userinfo/guide_tgpolicy.html)
- [12] R. Raman, and M. Livny, "Matchmaking: Distributed Resource Management for High Throughput Computing", in *Proc. of the 7<sup>th</sup> IEEE Symposium on High Performance Distributed Computing*, July 28031, 1998.
- [13] NSF TeraGrid Compute and Visualization Resources, <http://www.teragrid.org/userinfo/hardware/resources.php>
- [14] Condor DAGMan, <http://www.cs.wisc.edu/condor/dagman/>