Moving Volunteer Computing towards Knowledge-Constructed, Dynamically-Adaptive Modeling and Scheduling

M. Taufer[1], A. Kerstens[1], T. P. Estrada[1], D. A. Flores[1], R. Zamudio[1],
P. J. Teller[1], R. Armen[2], and C. L. Brooks III[2]

[1]University of Texas at El Paso
Dept. of Computer Science
El Paso, TX 79968 USA
{mtaufer, akerstens, tpestrada,
daflores, rzamudio, pteller}@utep.edu

[2]The Scripps Research Institute
Dept. of Molecular Biology
La Jolla, CA 92037 USA
{rarmen, brooks}@scripps.edu

Abstract

Volunteer computing projects supported by BOINC have been exploring new research directions. For example, mature projects like Folding@home are moving towards the use of a broader range of architectures and computers. Other projects such as Docking@Home are exploring multi-scale, resource-driven and application-driven adaptations of the volunteer system.

This paper presents results that enforce the need for knowledge-constructed capabilities in volunteer computing projects, i.e., the capability to drive simulations based on application-results and resource-status. The Docking@Home project, which uses volunteer resources to study putative drugs by computationally simulating the behavior of small molecules (ligands) when docking to a protein, serves as a case study to positively assess two key hypotheses. The first hypothesis claims that the adaptive selection of computational models for docking simulations based on the features of the protein and ligand can positively affect the final accuracy of the prediction. The second hypothesis claims that the adaptive selection of volunteer resources can ultimately improve project throughput.

1. Introduction

Volunteer Computing (VC) deploys computing resources (e.g., desktops and notebooks) owned by volunteers and connected through the Internet to address fundamental problems in science. BOINC (Berkeley Open Infrastructure for Network Computing) is a well-known representative of VC middleware. It supports VC projects in Biology, Medicine, Mathematics, Strategy Games, Astronomy, Physics, Chemistry, and Earth Sciences. Currently the computing power of BOINC is about 420 TeraFLOPS (based on credit granted across all projects). The total free disk space on computers running SETI@Home, one of the first VC projects, is approximately 12 Petabytes. This makes BOINC a very attractive solution for computational projects that need significant computational sampling.

VC projects supported by BOINC have been exploring new research directions. For example, mature projects like Folding@home are moving towards the use of a broader range of architectures and computers. Folding@home recently announced a BOINC client that runs on the Sony PlayStation 3, which has a peak floating-point speed of 100 GFLOPS. 10,000 of these machines would yield one PetaFLOP (quoted from CNN.com). Other projects such as Docking@Home explore multi-scale, resource-driven and application-driven adaptations of the system. Docking@Home, in particular, is a project that studies putative drugs by computationally simulating the behavior of small molecules, called ligands, when docking to a protein. Docking@Home has recently gone live and aims to:

1. model protein-ligand docking with algorithms that dynamically adapt to the complexity and characteristics of the proteins and ligands under investigation: in other words, selecting at runtime the most appropriate and effective computational model (e.g., protein and solvent representation) for a given protein-ligand complex;

2. assure that these algorithms can be executed in the ”required” amount of time using large numbers of distributed VC systems: in other words, selecting at run-

1-4244-0910-1/07/$20.00 ©2007 IEEE.

time the most suitable computational resources based on CPU speed, memory size, and network connection.

It is of vital importance that these two challenges are addressed in concert and not by themselves, because algorithmic complexity translates into increased simulation time and resource characteristics constrain project throughput. This paper presents results that substantiate and enforce the approach to meeting these challenges that is being pursued in Docking@Home. In particular, the paper discusses two main hypotheses that substantiate the need for knowledge-constructed capabilities, i.e., the capability to drive simulations based on application-results and resource-status.

- Hypothesis 1: Protein-ligand complexes are differentiated by distinctive characteristics (e.g., the size and flexibility of both the protein and ligand) and can be grouped in test-cases accordingly. The easiest test-case is docking a small rigid ligand with a few rotatable bonds to a relatively rigid protein binding site, while a difficult test-case is docking a large flexible ligand with many rotatable bonds to a very flexible protein binding site. Docking models of increasing complexity and computational expense are expected to provide sufficient accuracy to correctly predict difficult test-cases. It is known that simple computationally inexpensive docking models provide sufficient accuracy for simple test-cases, and that these models are not adequate for more difficult test-cases.

- Hypothesis 2: Most VC projects use simple scheduling policies based on First-Come-First-Serve protocols. However, some volunteer resources are more available and reliable than others. Use of scheduling policies based on dynamically changing availability and reliability thresholds, which take into account fluctuations of availability and reliability over the whole population of workers, should positively affect project throughput.

To prove our hypotheses we have greatly benefited from SimBA, the discrete event Simulator for BOINC Applications that we developed as part of the Docking@Home project and which we briefly describe in Section 2.2.

The remainder of this paper is structured as follows: Section 2 provides background on BOINC, the Simulator for BOINC Applications (SimBA), and the VC projects we focus on in this paper. Section 3 discusses application-result driven adaptation methods, while Section 4 shows resource-status driven adaptations. Last but not least, Section 5 concludes the paper.

2. Background and Related Work

This section describes the BOINC framework, the Simulator for BOINC Applications (SimBA), and several VC projects used to assess our hypotheses, i.e., Predictor@home (P@H), the projects in the World Community Grid (WCG), and Docking@Home (D@H).

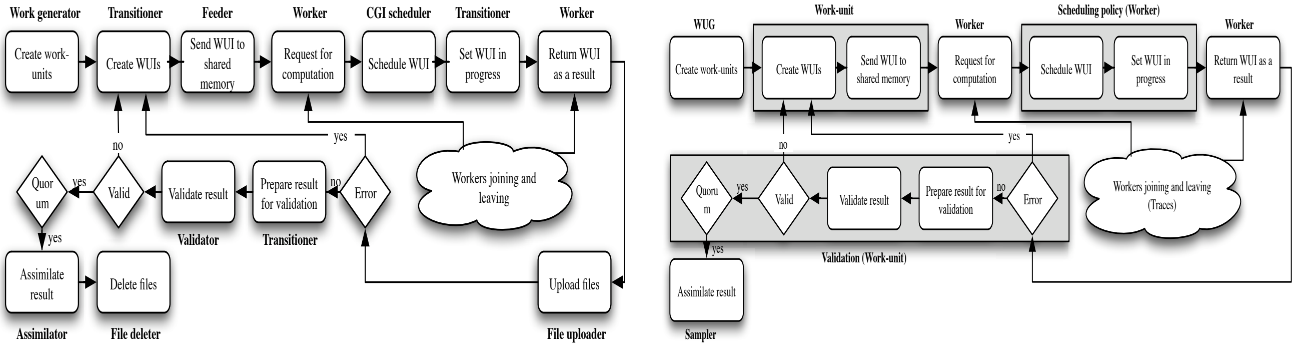
2.1 BOINC

BOINC (Berkeley Open Infrastructure for Network Computing) is a well-known representative of VC environments [1]. It is an open-source framework that harnesses the computing power and storage capacity of millions of PCs owned by the general public for large-scale scientific simulations. BOINC is based on the master-worker paradigm and uses replica computing to make sure that scientists can trust the final results returned by the volunteers. The BOINC server (master) does this by generating a specified number of replicas for a work-unit (WU) and distributing these to different volunteer computers (workers). When a worker successfully sends back a result, it is rewarded with credit (so-called Cobblestones) once the result has passed certain validation checks that aim to determine whether or not the result can be trusted. Validation is accomplished by comparing all of the returned replicated results and making sure at least a specified number of results are in agreement. If this is not the case, the BOINC master will send out new replicas until enough returned results are in agreement and a canonical result can be created. A result may be invalid as a result of malicious attacks, hardware malfunctions, or software modifications. As in other VC middleware packages, the computing resources available to a BOINC project are highly diverse: the hosts differ by orders of magnitude in their processor speed, available RAM, disk space, and network connection speed.

2.2 SimBA a Discrete Event Simulator for VC Projects

SimBA (Simulator of BOINC Applications) is a discrete event simulator we have developed that accurately models the main functions of BOINC: generation, distribution, and monitoring of WUs that are executed in a highly volatile, heterogeneous environment like Volunteer Computing (VC), as well as collecting and validating results of the executed WUs [2, 3]. The one-to-one correspondence between the functions of BOINC and those of SimBA are shown in Figure 1.

Figure 1.a shows the general path of a WU from its generation to the validation of its results on the BOINC master: boxes represent actions and a label over or under the box indicates the BOINC daemon that performs the associated



(a) BOINC

(b) SimBA

Figure 1. Mapping of the functions of BOINC and SimBA

action. Figure 1.b shows the same path in SimBA where the actions are events (boxes) driven by entities (the label over or under the boxes). More in particular, SimBA realistically simulates the creation, characterization, and termination of workers (volunteer computers) by using trace files obtained from real BOINC projects, e.g., Predictor@home [7]. A trace file contains information that characterizes workers, e.g., creation time, OS, and life span. Given a trace file, SimBA:

- generates WUs and creates from each a number of instances or replicas;
- distributes tasks according to worker requests and the selected scheduling policy;
- models worker volatility and heterogeneity using the input worker characterization;
- determines the status of a worker's returned results using the worker's error and success rates;
- determines the validity of successfully completed replicated tasks using the quorum, i.e., the required number of "agreeing" results; and
- computes the performance of the simulated VC project in terms of throughput.

Currently SimBA supports Homogeneous Redundancy [8], First-Come-First-Serve as well as fixed and variable threshold-based scheduling policies [2, 3]. Our experience to date indicates that SimBA is a reliable tool for performance prediction of VC projects. Preliminary results show that SimBA's predictions of P@H performance are within approximately 5% of the performance reported by this BOINC project [3].

2.3 Traces from Predictor@home and the World Community Grid

Predictor@home (P@H) is a BOINC project for large-scale protein structure prediction [7]. The protein structure prediction algorithm in P@H is a multi-step pipeline that includes: (a) a conformational search using a Monte Carlo simulated-annealing approach using MFold [6]; and (b) protein refinement, scoring, and clustering using the CHARMM Molecular Dynamics simulation package [5]. World Community Grid (WCG) is a project supported by IBM that makes grid technology available to the public and not-for-profit organizations. WCG currently supports several VC projects: Help Defeat Cancer, FightAIDS@Home and Human Proteome Folding [11]. In Section 4 we use traces from P@H MFold simulations as well as from projects associated with WCG to assess Hypothesis 2.

2.4 Docking@Home

Docking@Home is a VC project that aims to investigate putative drugs computationally by simulating the docking of small molecules, called ligands, to proteins and predicting the geometry of the protein-ligand complex. In vivo, when a ligand binds to a protein the functionality of the protein changes, which is the desired action of a drug. For example, an enzyme is a protein that catalyzes a biochemical reaction: a small molecule (ligand) binds to the protein, which converts the ligand into a product and releases it. A competitive inhibitor to this process is a small molecule that is structurally similar to the ligand but cannot be converted into a product, thus inhibiting enzyme activity. Many drugs are designed to inhibit the activity of a specific enzyme in this way. Another example is a receptor protein that recognizes a signal molecule, called an agonist, that binds to the protein and sends a biochemical signal. An antagonist is a small molecule (ligand) that may bind to the protein receptor but cannot trigger the signal, or blocks the binding of agonists, thus inhibiting receptor activity. Drugs that are targeted to such protein receptors aim to either increase or decrease the level of the signal that is sent.

Given a protein-ligand complex formed by a protein and a ligand, a WU in Docking@Home simulates the docking process of the ligand to the protein-binding site. This process consists of a sequence of independent trials [9]. Each trial begins with a randomly selected starting conformation of the ligand and proceeds by investigating the docking of a set of randomly selected orientations of this ligand conformation at a specific docking site of the protein (docking attempts). Protein-binding sites might be known a-priori through information available in a protein-ligand database such as LPDB [4] or have to be identified and investigated in preliminary simulations.

In Docking@Home, the CHARMM program is used to perform the docking attempts in terms of Molecular Dynamics (MD) simulations and minimizations [9, 12]. The docking model is characterized by three factors: protein-ligand representation, solvent representation, and sampling strategy. Different levels (scales) of complexity can be chosen for the three factors, constituting multi-scale modeling of the protein-ligand docking process. The protein-ligand representation spans scale from a rigid to a flexible representation of proteins. The solvent representation spans scale from a less accurate to a more accurate modeling of water treatment. The sampling strategy spans scale from a fixed to an adaptive sampling of the protein-ligand docking space (i.e., length of single docking trials and number of trials per protein-ligand complex). Figure 2 shows an example of a trial for the LPDB protein-ligand complex 1cnx in which different docking attempts of the ligand to the protein are performed.

The search for putative drugs (i.e., ligands that dock well in a protein) is a search in a large space of potential conformations and, therefore, is a very time- and compute-intensive process, which can significantly benefit from using VC resources. The conformational space of a single ligand is very large, and large-scale virtual screening efforts such as the ”Find-a-Drug” project aim to dock more than one million ligands to a single target protein.

3. Moving towards Application-Result Driven Simulations

3.1 Our Hypothesis

While searching for effective techniques to drive simulations based on application-results in Docking@Home, our hypothesis is that protein-ligand complexes are differentiated by distinctive characteristics (e.g., the size and flexibility of both the protein and ligand) and can be grouped in more or less complex test-cases accordingly. Docking models of increasing complexity and computational expense are expected to provide sufficient accuracy to correctly predict difficult test-cases but are too computationally expensive for simpler test-cases for which simple docking models already provide sufficient accuracy.

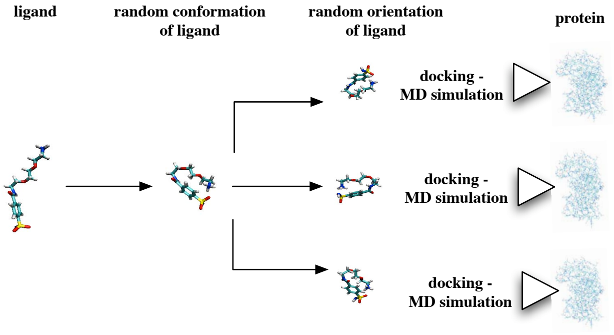


Figure 2. A docking trial in Docking@Home

3.2 Computational Assessment

To assess our hypothesis we ran the following computational test: we considered 26 protein-ligand complexes from the LPDB database [4] in which the ligands have different numbers of rotable bonds. Rotable bonds in a ligand represent its flexibility to change its conformation. We also considered three different docking models:

- Model 1: a docking protocol that employs a coarse representation of a protein, i.e., a lattice with 1 Å distance between two consecutive points and 10 docking trials per protein-ligand complex
- Model 2: a docking protocol that employs a coarse representation of a protein, i.e., a lattice with 1 Å distance between two consecutive points, but 20 docking trials per protein-ligand complex
- Model 3: a docking protocol that employs a finer representation of a protein, i.e., a lattice with 0.25 Å distance between two consecutive points and 40 docking trials per protein-ligand complex

Model 3 is considered only for ligands with more than three rotable bonds. We cluster complexes based on the number of rotable bonds and observe whether different models provide different results (with different accuracies) based on the complexity of the ligand: simple ligands are those with three or fewer rotable bonds, while complex ligands are those with more than three rotable bonds. Figure 3 presents the accuracy of the three docking models. For each model and associated docking attempts, best-docked structures are identified at runtime and are those with lowest energy. The accuracy of any given docking attempt is measured by the Root-Mean-Square-Deviation (RMSD) of all non-hydrogen ligand atoms between these lowest-energy structures from the docking attempt and the ligand's position in the crystal structure. Good structures are those with RMSDs smaller than two Å; however, structures with RMSDs between two and three Å are considered good results for difficult test-cases. As shown in the figure, only in six cases, over a total of 26, do we get an improvement in accuracy (RMSD decreases) by increasing the number of trials from 10 to 20. For ligands with a higher flexibility (number of rotable bonds greater than three), we consider the more accurate model (Model 3), with a finer lattice structure, which results in a larger memory footprint and increased execution time for each docking attempt. In these cases, we observe a significant improvement in accuracy for 12 of the 15 complexes. We do not apply the more sophisticated Model 3 to small and inflexible ligands because this level of complexity is not needed. Performance analysis of the different models is presented in [9].

3.3 Work in Progress

The results in Section 3.2 encourage us to pursue the identification of protein-ligand characteristics and the adaptation of docking models to the complex's characteristics. In particular, in Docking@Home we are extending the complexity of the models and the factors that characterize the protein-ligand docking process towards a multi-scale modeling representation that eventually will comprise three spanning scales (dimensions) of docking assumptions:

- Protein-ligand representation - from a rigid to a flexible protein representation.
- Solvent representation - from a constant dielectric coefficient to a distance-dependent dielectric coefficient and ultimately an implicit representation of water using a Generalized Born model.
- Sampling strategy - from a fixed number of trials per attempt and a fixed number of orientations per conformation to a variable number of trials per attempt and a variable number of orientations per conformation.

Selection of these spanning scales can be performed manually but this would result in time-demanding tuning of the model by the scientists for each protein-ligand complex. On the other hand, an automatic selection of models at runtime would facilitate the scientists' work. To drive the selection of the most suitable scales at runtime, for a given protein-ligand complex, we are studying the effectiveness and the cost of using machine learning techniques such as fast kernel classifiers, support vector machines, support vector regression, and active learning combining off-line evolutionary strategies. We plan to deliver results of this study in the second phase of the Docking@Home project.

4. Moving towards Resource-Status Driven Simulation

4.1 Our Hypothesis

While looking for effective solutions for resource-driven simulations, our hypothesis is that the use of scheduling policies based on dynamically changing availability and reliability thresholds, which take into account fluctuations of availability and reliability over the whole population of workers, should positively affect project throughput.

4.2 Computational Assessment

To assess this hypothesis some initial definitions are needed. First of all, metrics are needed to quantify the availability and reliability of volunteer computers. In [2, 10] we propose availability as the ratio of the number of work-unit (WU) replicas returned to the number of WU replicas assigned to a host. This ratio is measured over a certain interval of time or a certain number of bundles of replicas assigned to that host. As pointed out in [2, 10], hosts are error prone and might return fewer results than replicas assigned. Again, in [2, 10], we define reliability as the ratio of the number of valid results to the number of results returned without errors. A valid result is one that passed the validation check in a BOINC project, which can be either a fuzzy or a bit-to-bit comparison. Malicious attacks, hardware malfunctions, or software modifications are factors that could cause differences in successfully returned results. The two metrics that we use to quantify availability and reliability are only two of several possibilities. Developers of the World Community Grid are considering other metrics using claimed and granted credits [11]. To validate our hypothesis we use SimBA for the simulation of a VC project and traces from the Predictor@home (P@H) BOINC project. The simulated project has a minimum number of three replicas per WU and an estimated 30 GFLOPS per WU. There are 14,000 workers in the P@H traces and we simulate a project run of 12 days. Last but not least, we apply Homogeneous Redundancy (HR) [8] in our simulation and use a bit-to-bit validation of the results. HR uses the fact that replicas are assigned to computationally equivalent machines (e.g., only to Windows machines with Intel processors). In this scenario we consider three different scheduling policies:

- Policy 1: First-Come-First-Serve (FCFS) - replicas are assigned to any worker that asks for work (availability

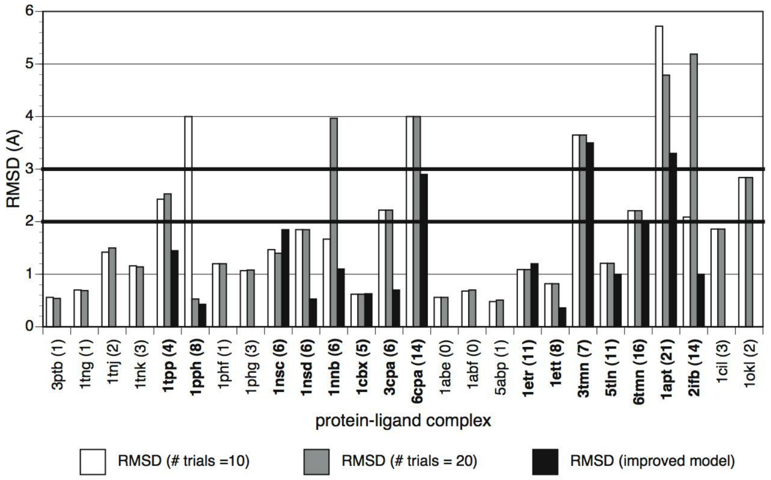


Figure 3. Comparison of docking accuracy for different protein-ligand complexes and models

and reliability thresholds are both zero)

- Policy 2: Fixed thresholds - replicas are assigned to workers that have an availability and a reliability above some fixed thresholds defined by the project administrator
- Policy 3: Dynamically adaptable thresholds - a heuristic is applied at a regular interval to vary the thresholds (ranging from .1 to 1) at runtime; if the number of replicas waiting for distribution is greater than the number of work-unit (WU) replica requests generated by ineligible hosts, i.e., hosts with reliability and availability levels below the current thresholds, then both the availability and reliability thresholds are decreased by 0.1 (10%), otherwise, they are increased by 0.1 (10%)

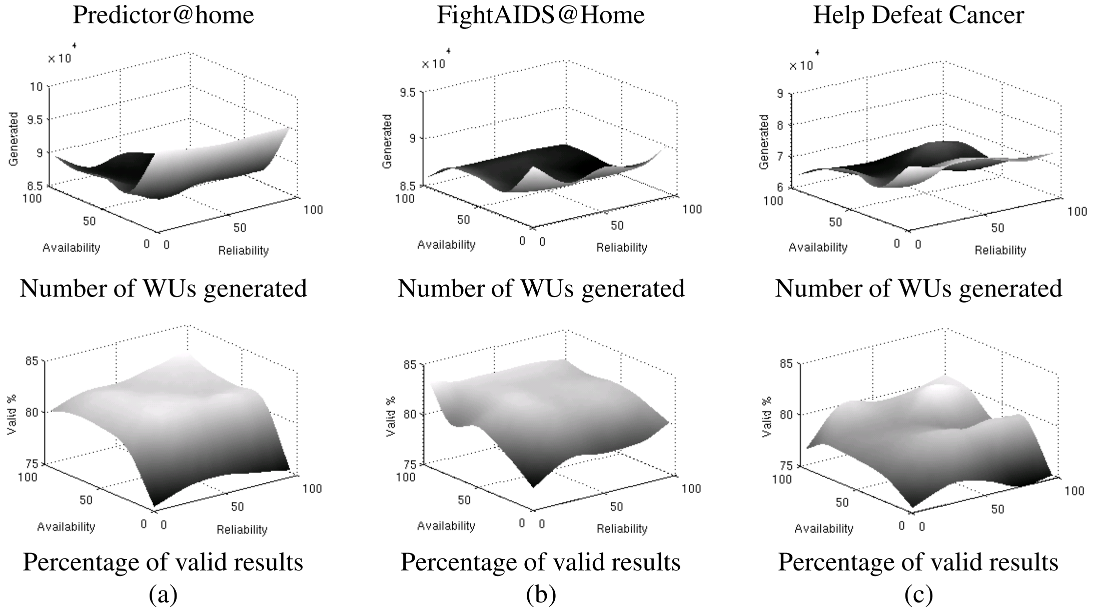
Table 1 presents the results of the related simulation. We correlate the results of the second and third policy to the first, the FCFS policy, which is the default scheduling policy in BOINC. As can be seen in the table, with the fixed threshold policy, which excludes those machines that have demonstrated poor availability or have returned only a few valid results, the number of generated WUs remains the same but the number of replicas that are distributed to volunteer computers decreases. Therefore, this reduction is not due to less sampling but to the significant reduction in errors (-64.1%) and the consequent reduction in production of additional replicas (-11.2%). The BOINC master keeps track of distributed replicas and assigns each WU replica a deadline for completion. If a WU replica does not complete within this defined time or if it fails, then the BOINC master classifies it as timed-out or error, respectively. Then, the master proceeds with the generation and distribution of a new replica of the WU until a specified quorum of WU replicas needed for validation of the returned results is attained. In P@H and SimBA the quorum dictates that three results must match; the maximum number of replicas per WU is five.

The number of average replicas per WU drops from 3.8 for FCFS to 3.2 for the fixed threshold policy. The ideal thresholds for the fixed threshold policy depend on the application and the available VC resources. By adopting dynamically adaptable thresholds that are modified in response to changes in VC resource availability, we observe in Table 1 that we get even better performance than using the fixed thresholds: the number of valid WU results increases by 2.8% and the average number of replicas per WU remains 3.2. In addition to improving project throughput, by varying the thresholds at runtime, we can regain previously ineligible volunteer computers and meet the needs of VC projects with diverse ideal thresholds. Figure 4 shows that, although different VC projects with different resources show similar tendencies as the availability and reliability thresholds increase, each VC project has its own sweet spot(s): the availability and reliability thresholds for which the percentage of valid results is the highest. The figure shows this behavior for three different VC projects: P@H, FightAIDS@Home, and the Help Defeat Cancer project. For each project we have used SimBA and the project's traces to generate the number of WUs and measure the percentage of these that are valid.

Table 1 and Figure 4 show that our hypothesis of adapting the threshold levels according to the resource-status by using simple heuristics results in better project throughput.

Table 1. P@H performance of different scheduling polices based on SimBA results

	FCFS	Fixed Thresholds (Best case: Avail. 95%; Rel. 75%)		Dynamically Adaptable Thresholds	
Generated WUs	78658	78886	+0.2%	78621	~0%
Generated WU Replicas	284140	252258	-11.2%	253611	-10.7%
Errors (Replica)	38491	13802	-64.1%	14424	-62.5%
Valid WUs	70948	71201	+0.4%	72929	+2.8%
Average Replicas per WU	3.8	3.2	-11.1%	3.2	-11.1%



Predictor@home

FightAIDS@Home

Help Defeat Cancer

Number of WUs generated

Number of WUs generated

Number of WUs generated

Percentage of valid results

Percentage of valid results

Percentage of valid results

(a)

(b)

(c)

Figure 4. Work-unit generated and their percentage of valid results for different VC projects

4.3 Work in Progress

Our work is moving towards the comparison and contrast of a wider range of metrics to effectively capture VC resource-status. Currently we are comparing credit-based metrics (claimed and granted credits) with our availability and reliability metrics. SimBA has been serving as a useful tool to study and quantify the effectiveness of new metrics as well as new scheduling techniques based on heuristics and machine learning techniques. As shown in Figure 4 (b) and (c), we have extended our database of traces to include traces from the World Community Grid project and we will soon have new traces from the Docking@Home project (currently in an alpha test phase). The quantification of the scheduling effectiveness of the different adaptive scheduling techniques in VC projects using SimBA will include the analysis of time to distribute WU replicas, the number of ineligible volunteer computers, and the throughput of valid results. Another critical aspect is how to use ineligible machines: we do not want to let these machines starve. We are currently studying new scheduling strategies that assign priorities to WU replicas and distribute them to hosts based on these priorities, e.g., high priority replicas are assigned to highly available and reliable machines and vice versa, low priority replicas are assigned to less available and less reliable machines. Ultimately, we will integrate the most effective scheduling techniques in Docking@Home.

5. Conclusion

Volunteer Computing (VC) environments are at the edge of the frontier of heterogeneous computing. Even so, they use very simple scheduling policies and application simulation protocols. As indicated by the results presented in the paper, dynamic adaptivity in terms of both application modeling and resource scheduling can provide a path to the next generation of VC. Our results point out the need for knowledge-constructed capabilities in VC projects: knowledge of application- and system characteristics should drive runtime adaptations of the generation and distribution of computation. The Docking@Home project is studying methods and strategies to build these knowledge-constructed capabilities and use them to adapt both application modeling and resource scheduling to ultimately achieve higher project throughput.

6. Acknowledgments

This work was supported by the National Science Foundation, grant # SCI-0506429, DAPLDS - a Dynamically Adaptive Protein-Ligand Docking System based on multiscale modeling. The authors want to thank Kevin Reed (IBM) and the World Community Grid for the traces used in SimBA to build Figure 4 (b) and (c). We thank David Anderson and the BOINC community for their time, dedication, and computer resources.

References

[1] D. P. Anderson. Boinc: A system for public-resource computing and storage. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing.*, pages 4–10, November 2004.

[2] T. Estrada, D. Flores, M. Taufer, P. J. Teller, A. Kerstens, and D. P. Anderson. The effectiveness of threshold-based scheduling policies in boinc projects. In *Proceedings of the 2nd IEEE International Conference on e-Science and Grid Technologies (eScience 2006)*, December 2006.

[3] D. Flores, T. Estrada, M. Taufer, P. J. Teller, and A. Kerstens. Simba: a discrete event simulator for performance prediction of volunteer computing projects. In *Proceedings of SC2006, ACM/IEEE Supercomputing 2006*, November 2006. Poster.

[4] LPDB - protein-ligand database. http://lpdb.scripps.edu/.

[5] A. D. MacKerell Jr., B. Brooks, C. L. Brooks III, L. Nilsson, B. Roux, Y. Won, and M. Karplus. *CHARMM: The Energy Function and Its Parametrization with an Overview of the Program*, volume 1. John Wiley & Sons, 1998.

[6] J. Skolnick, A. Kolinski, and A. R. Ortiz. Monsster: a method for folding globular proteins with a small number of distance restraints. *Journal of Molecular Biology*, 265(2):217–241, January 1997.

[7] M. Taufer, C. An, A. Kerstens, and C. L. Brooks III. Predictor@home: A protein structure prediction supercomputer based on global computing. *IEEE Transactions on Parallel and Distributed Systems*, 17(8):786–796, August 2006.

[8] M. Taufer, D. P. Anderson, P. Cicotti, and C. L. Brooks III. Homogeneous technique to ensure integrity of molecular simulation results using public resources. In *Proceedings of the 14th Heterogeneous Computing Workshop (HCW 2005), IPDPS 2005*, April 2005.

[9] M. Taufer, M. Crowley, D. Price, A. Chien, and C. L. Brooks III. Study of an accurate and fast protein-ligand docking algorithm based on molecular dynamics. *Concurrency and Computation: Practice and Experience*, 17(14):1627–1641, December 2005.

[10] M. Taufer, P. J. Teller, D. P. Anderson, and C. L. Brooks III. Metrics for effective resource management in global computing environments. In *Proceedings of the 1st IEEE International Conference on e-Science and Grid Technologies (eScience 2005)*, December 2005.

[11] World Community Grid. http://www.worldcommunitygrid.org/.

[12] G. Wu, D. H. Robertson, C. L. Brooks III, and M. Veith. Detailed analysis of grid-based molecular docking: A case study of cdocker-a charmm-based md docking algorithm. *Journal of Computational Chemistry*, 24:1549–1562, 2003.