
**Docking@Home Newsletter** **19-Dec-06 Issue #1**

Welcome to the first issue of the Docking@Home newsletter that you will hopefully see regularly in your email inbox. It is meant to inform you of the status of the (sub)projects the D@H team is working on.

# Table of Contents



## 1 The Project

As most of you know, the Docking@Home project is the brain-child of Michela Taufer, currently an assistant professor in the Computer Science department of the University of Texas at El Paso (UTEP). Docking@Home is the implementation of the Dynamically Adaptive Protein-Ligand Docking System (DAPLDS) project which involves collaborations among the University of Texas at El Paso (UTEP), The Scripps Research Institute (TSRI), and the University of California at Berkeley. This project enables adaptive multi-scale modeling in a volunteer computing environment and will further the knowledge of atomic details of protein-ligand interactions. By doing so, it will accelerate the discovery of novel pharmaceuticals. The goals of the project are: (1) to explore the multi-scale nature of algorithmic adaptations in protein-ligand docking and (2) to develop cyber-infrastructure based on computational methods and models that efficiently accommodate these adaptations.

Docking@Home Newsletter 19-Dec-06 Issue #1

## 2 The Team

The Docking@Home project team consists of a group of faculty, researchers and students. Here is the list:

**Principal Investigator (PI)** - Michela Taufer is an assistant professor in the Department of Computer Science at the University of Texas at El Paso (UTEP) and the leader of the Docking@Home project. Michela joined UTEP in January 2005. Her research interests include new algorithms and architectures for resource- and time-expensive applications in computational chemistry, physics, and biology; effective migration of large scale simulations to global computing systems based on public resources; and performance analysis, modeling and optimization of large-scale simulations on heterogeneous, distributed systems.

**Co-PI** - Patricia Teller is a professor in the Department of Computer Science at UTEP. Pat's research involves dynamic adaptation of applications, operating systems, and computer architectures; performance evaluation, modeling, and enhancements; parallel and distributed computing, computer architecture, operating systems, and simulation methodologies; and workload characterization. In the Docking@Home project her foci are SimBA, the simulator for BOINC applications; computational consistency; and task scheduling.

**Co-PI** - Martine Ceberio is an assistant professor in the Department of Computer Science at UTEP. Her main areas of interest are constrained global optimization, constraint solving, applications of interval analysis, and soft constraints.

**Co-PI** - Charles Brooks III has been a professor at the Scripps Research Institute (TSRI) since 1994. His main areas of interest are the application of statistical mechanics, quantum chemistry, and computational methods to chemically- and physically-oriented problems in biology.

**Co-PI** - David Anderson (well known by everybody in the volunteer computing community) is a research scientist, principal investigator, and director of the University of California at Berkeley's BOINC project and SETI@home project. His research interests include distributed systems, realtime and multimedia systems, graphics, computer music, communication protocols, and psychometrics applied to learning and aesthetic preference.

# Docking@Home

## Docking@Home Newsletter 19-Dec-06 Issue #1

**Senior Researcher** - Andre is a senior researcher and the HPC research computing coordinator at UTEP. Research-wise, he is working on the Docking@Home and DAiSES projects. In the Docking@Home project he is focusing on the BOINC simulator, the server daemons, and scheduler; he also is adapting the CHARMM application for the BOINC platform and is responsible for the Docking@Home computing infrastructure.

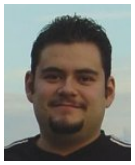
**Postdoctoral Associate** - Roger Armen is our project scientist and an NIH-funded postdoctoral fellow in the group of Charles Brooks III at TSRI. He specializes in the development and application of molecular docking techniques for the Docking@Home project.

**PhD Student** - Richard Zamudio is a PhD student in the Global Computing Lab. His research interests focus on developing adaptive scheduling techniques for Docking@Home. He also is looking into the computational differences associated with different architectures that we are experiencing with Charmm.

**PhD Student** - Trilce Estrada is a PhD student in the Global Computing Lab. Her research interests include Machine Learning (ML) applied to Bioinformatics and High Performance Computing, as well as the design of new ML algorithms. In the Docking@Home project she is currently working on the design and implementation of new scheduling algorithms in SimBA.

**Masters Student** - David Flores is a Masters Student in the Global Computing Lab. His main work involves designing, implementing, and validating SimBA. This simulator is being used within the project to evaluate dynamic adaptation and scheduling techniques.

**Undergraduate Student** - Karina Escapita is an Undergraduate in the Global Computing Lab. Her main duties are maintaining the Docking@Home web site and designing a nice screen saver for the application.



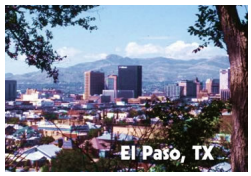
**Undergraduate Student** - Guillermo Lopez (we call him Memo) is an Undergraduate in the Global Computing Lab with healthy interest in performance analysis and distributed systems. Within the Docking@Home project, he is researching divergences caused by floating point implementations on different architectures. He can also be found on the D@H forums and is an eager alpha cruncher. He will start his Masters studies in the Spring of 2007.

# Docking@Home

**Docking@Home Newsletter** **19-Dec-06 Issue #1**

**Our Forum Moderators** - Since the start of the Docking@Home project we have found two people that agreed to be moderators on our forums: Suguruhirara and David Ball. They already have gathered a lot of knowledge on D@H and will happily answer any questions or resolve any issues you have. You are doing a great job guys!

## 3 The City





El Paso is located in the Chihuahuan Desert of extreme western Texas, along the Rio Grande River. It adjoins both the state of New Mexico and the country of Mexico with the Franklin Mountains, the southern tip of the Rockies, slicing El Paso nearly in two.

With its classic Western geography and because it shares an international border with Ciudad Juarez, Mexico's rich culture pervades everything in El Paso, from its art and architecture to its celebrations and cuisine. El Paso's area is 248 square miles, making it the fourth largest city in Texas, and 22nd in size in the United States. It is the nation's third fastest growing metropolitan area. El Paso is midway between Los Angeles and Houston and resides in the Mountain Time Zone.

## 4 Project Status

So what is the current status of D@H and where are we going? Currently the following sub-projects are ongoing in the group:

**a) Running Charmm on architectural different platforms** - Memo, Richard, Andre, Michela and Pat are focusing on experiments that will give them enough information to ensure that when Charmm runs on different architectures or, at least a set of architectures, the same results will be generated. This is a very challenging problem and has turned out to eat up a lot of our time. It is also the reason we employ Homogeneous Redundancy (HR) on BOINC to run our project. HR ensures that results within a workunit will only be executed on architecturally equivalent platforms. We currently have 6 groups of these:

- Windows on AMD/PII/PIII
- Windows on Intel (non-PII/III)
- Linux on AMD/PII/PIII
- Linux on Intel (non-PII/III)
- Mac Intel

- Mac PPC (no app version yet)

There are many strategies to find out the source of the differences and how to solve this problem. One is to experiment with the many compiler options available for floating point operations. Memo is currently experimenting with this to see if there is light at the end of the tunnel. Michela has visited the group that runs Leiden Classical in Holland who appear to be experiencing the same type of problems that we are having; they have solved this by introducing many more HR classes and dynamically partitioning the BOINC shared memory segment for these classes. This might be a solution that could fit Docking@Home too. We are currently looking into this.

**b) Linux ulimit fix** - We have noticed that on some Linux distributions it is necessary to increase the stack size limit to unlimited, because if the stack size is set too low, the application crashes with the following error: 'Charmm exited with code 1'. We have determined that distributions based on RedHat and Debian need this fix. SuSE, Slackware and Mandrake-based distributions have the stack size set to unlimited out of the box. Here are some instructions on how to implement the workaround: On the Bash and K shells your setting can be seen by typing 'ulimit -s' in a terminal. With TC shell the command is 'limit'. To find out which shell you are running enter the command 'echo $SHELL'. To make the Charmm 'exit 1' errors go away, please set the stack size to unlimited using the command 'ulimit -s unlimited' (or 'limit stacksize unlimited' for TC shell users) before you start the BOINC manager or add it to your run_manager.sh and/or run_client.sh start scripts in the BOINC installation directory. For people who download BOINC from their distribution's repository, please add it to the BOINC start script that will most likely be located in the directory /etc/init.d. For example, on Ubuntu the line 'ulimit -s unlimited' has to be added to /etc/init.d/boinc-client. We will have to find a server-side solution for this stack size problem before we can go to Beta and Production.

**c) Checkpointing** - Many of you have experienced a lot of disk activity when Charmm is running on your computers. The reason for this is that we currently write a lot of debug information to the Charmm logfile charmm.out. This of course is a cause for lots of disk writing; since we are in alpha we will have to do this to find out problems more easily. as soon as we have most of our pressing problems solved, we can cut back on the writing of extensive information. Another problem is that we currently do not respect the disk writing user preference: for example, many users have set that BOINC applications should write to their disk not more than every 60 seconds. The

function call that applications should use to find out this information has not been integrated into Charmm yet. We will tackle this issue as soon as Charmm c33b1 is released by the Charmm developers at TSRI. Yet another problem with checkpointing is the fact that our checkpoints are currently not atomic and thus are sometimes interrupted by the scheduler in the BOINC client or a system crash or freeze (which seems to happen frequently on Windows machines). Use of the BOINC function boinc_time_to_checkpoint() will more than likely solve this problem.

**d) New Charmm version c33b1** - The new Charmm version is currently extensively being tested by the developers at TSRI. Charmm contains many unit test functions that should pass before a new version is released to the community. In the case of the BOINC functionality integrated into Charmm the unit tests have not all succeeded yet and there are some problems compiling Charmm for the Mac. When all tests pass and the Mac compile is successful, the new version will be released to the Docking@Home team. We already have a pre-released, non-production and BOINC-less version of c33b1 to play with, but this cannot be used on the project server.

**e) Credit granting methodology** - There are many problems with cross-platform credit granting. Since Charmm runs much better (read: faster) on Linux and Mac than it does on Windows, there are issues with the credit that a host claims for the performed work. On top of this, the BOINC client for Linux supposedly reports much lower benchmarks than it would do when the same host runs Windows. This is due to the fact that the BOINC client on Linux has not been compiled with all the optimizations in place like the Windows client has. This will highly likely be corrected in the next release of the BOINC client. From a general point of view, the credit claiming/granting problem can probably be best solved by assigning a fixed amount of credit based on the work performed. For example, this would mean that workunit A (needing 9 billion FLOPs or 9 gigaFLOPs) will for example grant a host 10 credits, while workunit B (needing 18 billion FLOPs or 18 gigaFLOPs) will grant 20 credits. That way the credit system is independent of the BOINC benchmarking system and thus more fair for our volunteers. In case you don't know what FLOP means: it stands for FLoating point OPeration and is a measure for a computer's performance.

**f) D@H Web Site** - Karina and Andre have made many improvements and fixes to the Docking@Home web site. One of the nicest changes to the web site is our new header, which was designed by two of our volunteers: Miss Atomic Booty and Miss Cori. Thanks again for this nice piece of work! There

are a couple of problems in the pipeline, one being the possibility to report offensive forum posts (the small red square with a cross underneath posts). Karina is working to find a fix for this bug.

**g) Wrapper and error handling** - Richard is working on extending the wrapper we implemented for Charmm with more sophisticated error handling routines. The goal is to return high-level, easy understandable information back to the scientists in case Charmm experiences a problem.

**h) Charmm crashes on Windows 98, ME and Millennium** - The aforementioned platforms experience a Charmm crash right after starting. This has to do with the fact that the Charmm logfile, charmm.out, cannot be opened. We have not found out yet why this is happening on these platforms. Richard is researching this issue.

**i) Screensaver Graphics** - Karina has started to work on some screensaver graphics for the Charmm application. Since she has just passed her Computer Graphics class, we thought she would be the best choice for the job! The first version of the screensaver will likely be a simple graphic bouncing around on the screen; later we plan to let it do more sophisticated things like showing progress, score and protein-ligand docking results.

**j) SimBA** - Our simulator for BOINC applications is continuously under development by Trilce, David F., Andre, Michela and Pat. We have started collaborations with World Community Grid and it is likely that a collaboration with the Leiden Classical group will appear as well. New scheduling algorithms are designed and tested using SimBA and the most promising ones might end up in the BOINC framework.


**Docking@Home Newsletter** **19-Dec-06 Issue #1**

# 5 Science Status

## *5.1 Introduction to Enzymes and Drug Targets.*

Proteins have various functions in the molecular biology of cells. An example of different types of protein functions are:

1. **structural proteins** - proteins that serve as structural building blocks or a scaffold for larger structures inside and outside of cells.
2. **transport proteins** - proteins that function by transporting biochemical cargo such as a hormone or a signal molecule from one location to another.
3. **protein receptors** - proteins on the cell surface that recognize biochemical signals such as neurotransmitters or signals from the immune system.
4. **enzymes** - these proteins participate in active biochemistry, they function by catalyzing a specific biochemical reaction and convert reactants to products. Enzymes are part of complex biochemical pathways. Some enzymes serve as steps in sequential pathways, and other enzymes function to regulate pathways.

**Enzymes** are one of the most important functional classes of proteins for drug design. If an enzyme converts a biochemical reactant (A) to a product (B), it is possible to design an **enzyme inhibitor** that is similar to the reactant (A) but does not allow the reaction to convert the inhibitor to the product (B). Enzyme inhibitors prevent or reduce the formation of product (B). Many common drugs marketed today function in this generic way as enzyme inhibitors. Most enzymes participate in complex biochemical pathways, and certain enzymes can be identified from disease research as **drug targets**. If research identifies a specific enzyme as being crucial to the progress of disease, then if this enzyme is targeted with a designed inhibitor, it may slow or reverse the progress of the disease. A large number of inhibitors are designed, synthesized, and evaluated for effectiveness. Usually only a single inhibitor with the most favorable properties and the least toxicity is selected out of thousands as a drug candidate and is then moved into clinical trials with patients.

For any given specific disease, scientists must first understand the function of specific proteins that are implicated in the pathogenesis of disease. Once these biochemical pathways are identified, then specific proteins may be

selected to be **drug targets**. If a specific disease is due to infection from a pathogenic agent (virus, bacteria, paracite) then the molecular biology of the life cycle of the pathogen and how it infects the body must be understood. Other human diseases are not due to infection from a pathogenic agent and are rather due to malfunctioning proteins or signaling pathways. Examples of diseases like these are cancer, autoimmune diseases, metabolic diseases, and neurodegenerative diesease such as Alzheimers and Parkinsons Disease.

***Docking@Home Drug Target Enzyme Classes***

One of the primary scientific goals of the DAPLDS project is to develop, improve and validate our CHARMM-based protein-ligand docking methods. To achieve this goal, we wish to focus on model protein-ligand systems with sufficient experimental data and with large biomedical impact. Therefore we aim to focus our initial studies to important classes of drug target enzymes and their inhibitors. Our constantly expanding knowledge of different types of inhibitors for specific enzyme classes will always be important for the design of new drugs. The study of well characterized drug targets remains very useful for improving methodology, which can then be applied to the design of new and improved drugs. Even more important is the fact that new drug targets are identified all of the time.

**Proteases** are enzymes that perform their function by cleaving a specific recognition sequence in the peptide chain of another protein. The common analogy is that proteases work like a pair of molecular scissors that cut proteins at specific locations. Many important biochemical pathways are regulated by protease activity. New protease drug targets are being identified all the time, especially from viral pathogen genomes.

**Aspartic proteases** became an important class of enzymes when HIV protease was identified as a drug target from the HIV genome. Inhibiting the HIV protease enzyme inhibits the replication of the virus and the spread of the infection to new healthy cells. The FDA approval of several potent HIV protease inhibitors such as saquinavir, ritonavir, indinavir, and nefinavir (to name a few) has revolutionized the treatment of HIV/AIDS.

Initially, we will study three aspartic proteases: **HIV protease, endothiapepsin, and penicillopepsin**. All three of these are aspartic proteases with many known specific inhibitors.

**Serine proteases** have always been an important class of protease enzyme

and are by far the most well studied of the proteases. Initially we will study three serine proteases: **α-thrombin, trypsin, and factor Xa**. All of these are well studied and are structurally characterized. Trypsin is a model system and not a drug target, but α-thrombin and factor Xa are drug targets for blood clotting disorders. Another newly discovered drug target in this protease class is Hepatitis C virus protease (NS3 protein). Up to 3% of the global human population may be infected with the Hepatitis C virus (HCV) which leads to cirrhosis of the liver and liver cancer. Hepatitis C viral protease NS3 will be the focus of future work in the later stages of the DAPLDS project.

A **kinase** is a class of enzyme that transfers a phosphate group from donor Adenosine 5'-triphosphate (ATP) molecule to a specific target molecule. In biochemistry this process is called phosphorylation. Kinases are a very important class of proteins that regulate many signal pathways via phosphorylation. Kinases have been identified as a very promising family of drug targets for cancer and other diseases such as autoimmune diseases. The first tyrosine kinase inhibitor Gleevec was recently approved by the FDA for treating cancer, and represents the potential of using kinase inhibitors for treating cancer. Specific kinase inhibitors work differently than other chemotherapy agents that non-specifically targets rapidly dividing cells. Gleevec is effective in treating chronic myelogenous leukemia, gastrointestinal stromal tumors, and other cancers. Kinases look like very promising drug targets, but one of the difficulties in designing kinase inhibitors is that many inhibitors are not specific enough for the target and may bind to many other members of the kinase family causing unintended effects. For this reason there is a lot of interest in detailed studies of kinase protein-ligand interactions for inhibitor design. Our initial studies of kinases will be focused on Cyclin-Dependent kinase 2 (**CDK2**), **cAMP kinase** (also protein kinase A, PKA), and **MAP kinase** (also known as P38). CDK2 is a model system and is one of the best characterized kinases that control the cell cycle. A closely related homolouge CDK4 is a drug target for cancer. cAMP kinase is a model kinase system, and its closely related homolouge PKB-α is a drug target for cancer. MAP kinase is also a model system and a drug target for cancer as well as autoimunne diseases including: rheumatoid arthritis, osteoarthritis, and inflammatory bowel disease. In the later stages of the DAPLDS project we will focus on extending our studies to many other members of the kinase family. New kinase drug targets are being identified and studied all the time.


**Docking@Home Newsletter** **19-Dec-06 Issue #1**

***Recent work on these Drug Target Enzyme Classes***

A lot of work goes in to preparing a protein-ligand complex to be ready for docking. In order to perform our CHARMM-based molecular docking calculations on a protein-ligand complex, one of the most challenging preliminary steps is to develop a reasonable potential energy function for both the protein and the ligand. Over the years much work has been done by many researchers in the CHARMM community to develop and verify various all-atom potential functions for proteins. However, much work still needs to be done to develop, improve, and verify a generalized potential function for small molecule ligands that is compatible with the CHARMM potential function for proteins. Changes to the potential energy function for the small molecule ligands and protein will effect the accuracy of docking results. Therefore we can validate changes to the potential function by docking accuracy.

The most challenging aspect of developing all-atom potential functions for small molecules is the incredible chemical diversity that is possible from combinations of various functional groups from organic chemistry (e.g. alkanes, alkyl halides, alchohols, thiols, ketones, ethers, esters, etc.). This results in a very large number of possibilities for novel small molecule compounds with diverse connectivities for these various functional groups. New ligands that have never been studied before may exhibit novel connectivity between functional groups that is not adequately described by the current potential function. Therefore, we must constantly expand and improve our small molecule potential function so it covers more diverse combinations of functional groups.

In the last two months we have been working on extending our small molecule potential function to cover many inhibitor types for the enzyme classes discussed previously: aspartic proteases, serine proteases, and kinases. We will soon be able to perform new docking calculations and cross-docking studies of:

- Aspartic proteases: HIV protease, endothiapepsin, and penicillopepsin
- Serine proteases: α-thrombin, trypsin, and factor Xa
- Kinases: CDK2, cAMP kinase, and MAP kinase


**Docking@Home Newsletter** **19-Dec-06 Issue #1**

## 6 Publications

The following publications/posters were a result of work done in the Docking@Home project:

- M. Taufer, A. Kerstens, T. Estrada, D. A. Flores, P. Teller, "SimBA: a Discrete Event Simulator for Performance Prediction of Volunteer Computing Projects", To appear in Proceedings of The International Workshop on Principles of Advanced and Distributed Simulation 2007, San Diego, California, June 2007
- M. Taufer, A. Kerstens, T. Estrada, D. A. Flores, R. Zamudio, P. Teller, R. Armen, C. L. Brooks III, "Moving Volunteer Computing towards Knowledge-Constructed, Dynamically-Adaptive Modeling and Scheduling", To appear in Proceedings of The Workshop on Large-Scale, Volatile Desktop Grids 2007, Long Beach, California, March 2007
- T. Estrada, D. A. Flores, M. Taufer, P. Teller, A. Kerstens, and D. Anderson, "The Effectiveness of Threshold-based Scheduling Policies", Proceedings of the 2nd IEEE International Conference on e-Science and Grid Technologies (eScience 2006), Amsterdam, Netherlands, December 2006
- D. A. Flores, T. Estrada, M. Taufer, P. Teller, and A. Kerstens, "SimBA: a Discrete Event Simulator for Performance Prediction of Volunteer Computing Projects", Poster in Proceedings of SC2006, ACM/IEEE Supercomputing 06, Tampa, Florida, November 2006


**Docking@Home Newsletter** **19-Dec-06 Issue #1**

## 7 Some Interesting Statistics

Here are some details gathered from the Docking@Home server.

**Volunteers** – 336

**Active volunteers (total credit > 0)** – 296

**Hosts** – 1118

- Windows – 856
  - Intel – 579
  - AMD – 277
- Linux – 218
  - Intel – 151
  - AMD – 67
- Macintosh – 44
  - Intel – 29
  - PPC – 15

**Active hosts (total credit > 0)** – 761

**Generated workunits** – 17000

**Valid workunits** – 9594

**Generated results** – 60224

**Valid results** – 25617

**Total credit** – 1,177,131

**Total Macintosh credit** – 81,325

**Total Windows credit** – 979,818

**Total Linux credit** – 115,988

Docking@Home Newsletter 19-Dec-06 Issue #1

## 8 What will the Future bring?

As you have seen in the previous sections, lots is going on in the project and we could not have done it without your help. When we have resolved most of our pressing problems (release Charmm c33b1, architectural differences, checkpointing, credit granting) we will go to the next Beta phase and open up the project for more volunteers.

Many thanks from everybody in the Docking@Home project team!

**Contact us at: dockingadmin@utep.edu**