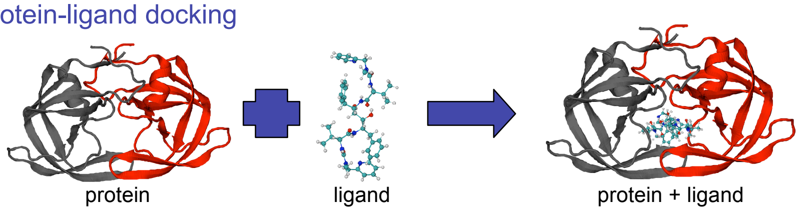
Volunteer Computing and Protein-Ligand Docking

Michela Taufer

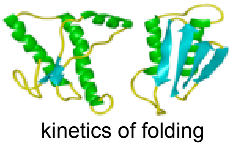
Global Computing Lab
University of Delaware

Simulating Biological Systems

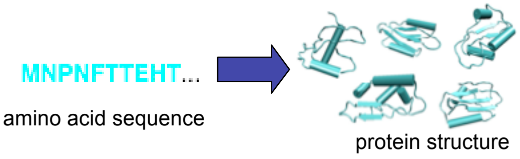
Protein-ligand docking



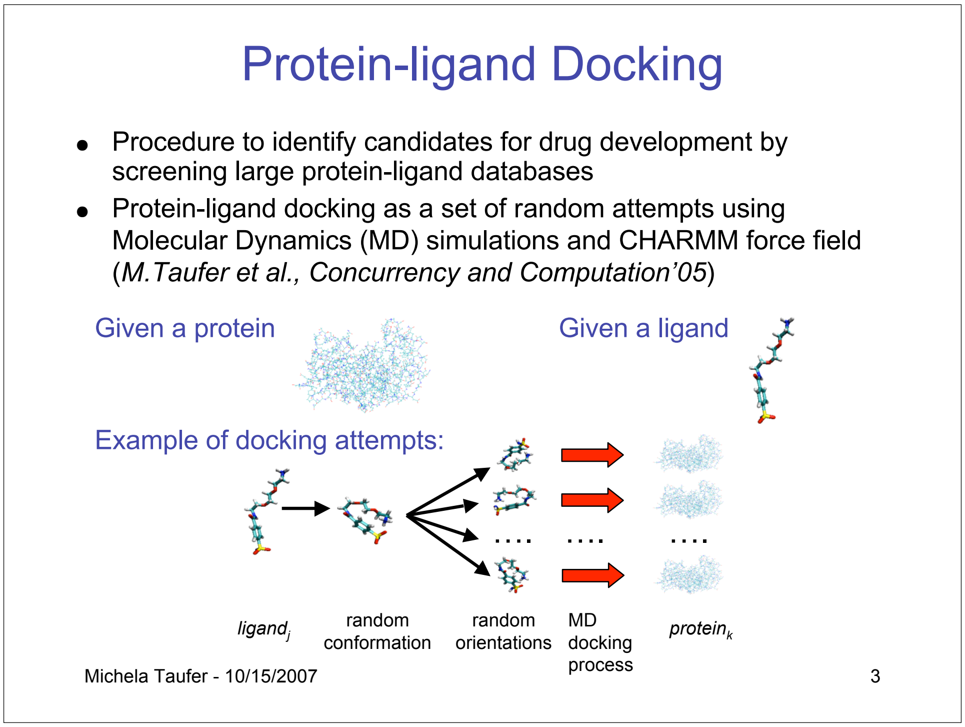
Protein folding



Protein structure prediction



- Search in large conformational spaces based on e.g., Monte Carlo and/or Molecular Dynamics simulations
- Different models of the biological systems require different amounts of resources/time and provide different levels of accuracy



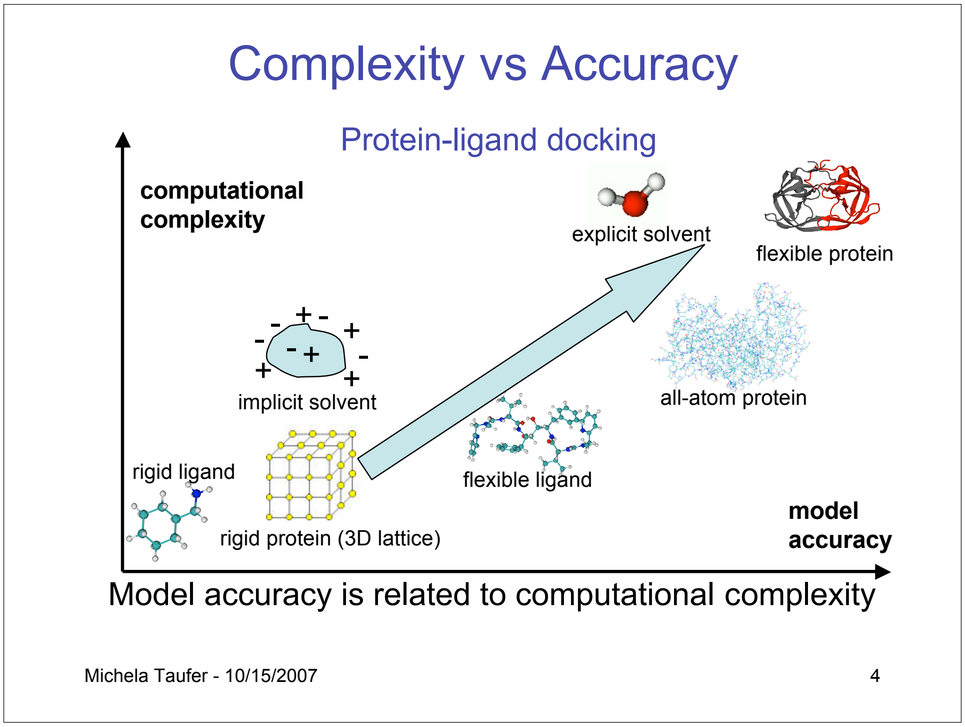
Protein-ligand Docking

- Procedure to identify candidates for drug development by screening large protein-ligand databases
- Protein-ligand docking as a set of random attempts using Molecular Dynamics (MD) simulations and CHARMM force field (*M.Taufer et al., Concurrency and Computation'05*)

Given a protein

Given a ligand

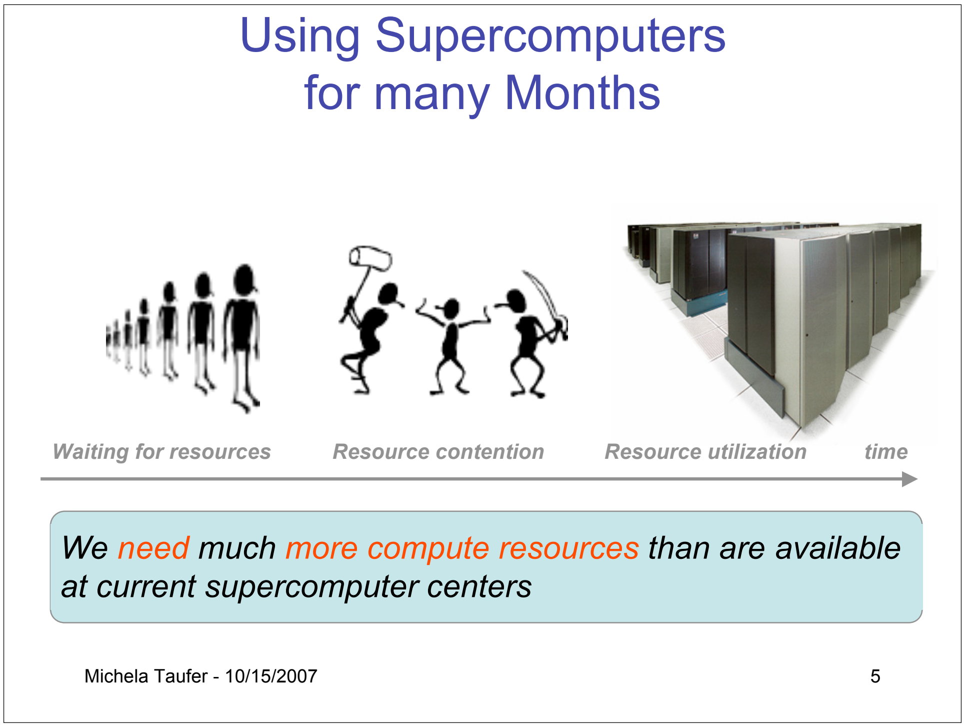
Example of docking attempts:



Complexity vs Accuracy

Protein-ligand docking

Model accuracy is related to computational complexity



Using Supercomputers for many Months

Waiting for resources

Resource contention

Resource utilization

time

We need much more compute resources than are available at current supercomputer centers

Searching for New Compute Resources: Volunteer Computing

Late 80s

Middle 90s

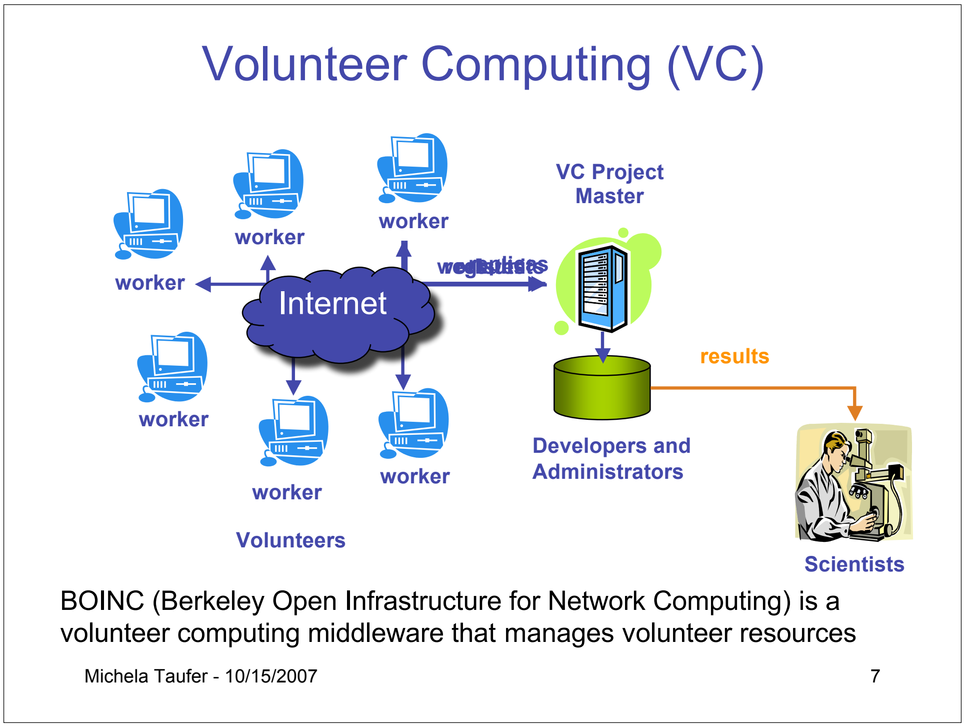
2000

time

Cray C90

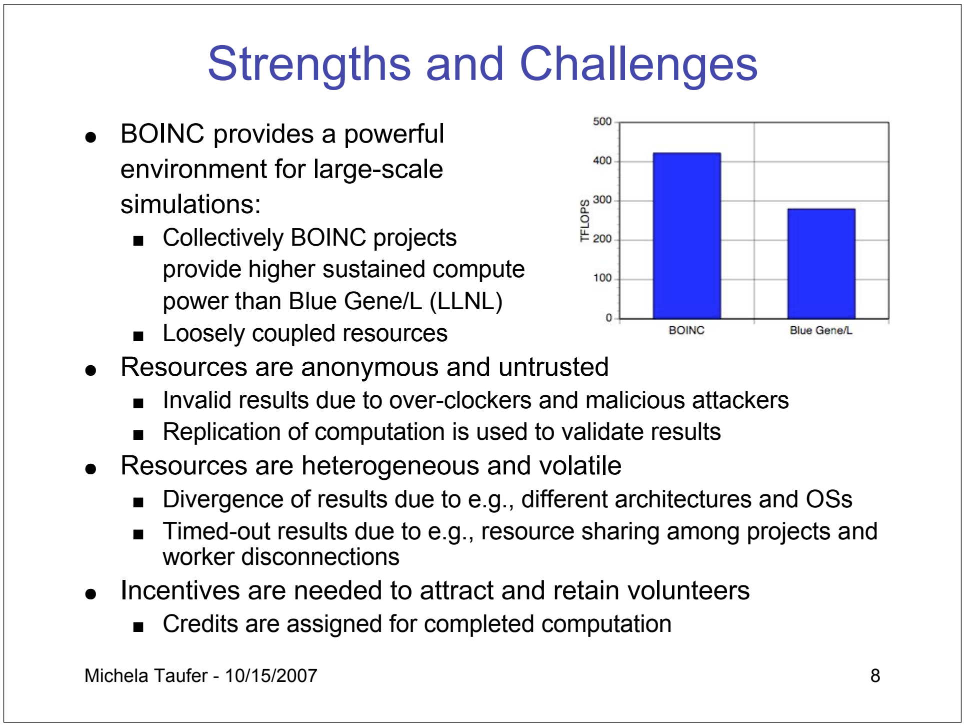
Cluster of PCs

Volunteer computing



Volunteer Computing (VC)

BOINC (Berkeley Open Infrastructure for Network Computing) is a volunteer computing middleware that manages volunteer resources



Strengths and Challenges

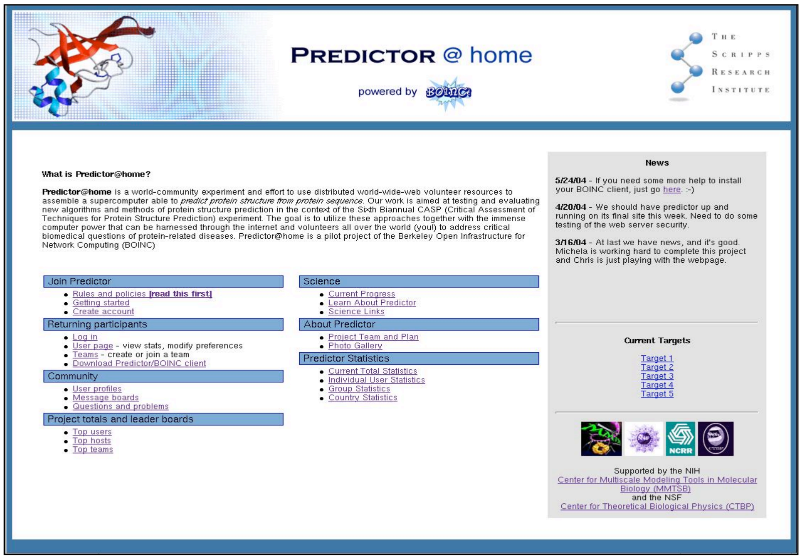
- BOINC provides a powerful environment for large-scale simulations:
  - Collectively BOINC projects provide higher sustained compute power than Blue Gene/L (LLNL)
  - Loosely coupled resources
- Resources are anonymous and untrusted
  - Invalid results due to over-clockers and malicious attackers
  - Replication of computation is used to validate results
- Resources are heterogeneous and volatile
  - Divergence of results due to e.g., different architectures and OSs
  - Timed-out results due to e.g., resource sharing among projects and worker disconnections
- Incentives are needed to attract and retain volunteers
  - Credits are assigned for completed computation

VC Projects

- Predictor@home (P@h)
  - TSRI; protein structure prediction
- Climateprediction.net
  - Oxford,UK; global climate change study
- Rosetta@home
  - U. Washington; protein study
- SETI@home
  - U.C. Berkeley; SETI
- World Community Grid
  - IBM; several life science applications

...and about 30 others

Predictor@home



What is Predictor@home?

Predictor@home is a world-community experiment and effort to use distributed world-wide-web volunteer resources to assemble a supercomputer able to *predict protein structure from protein sequence*. Our work is aimed at testing and evaluating new algorithms and methods of protein structure prediction in the context of the Sixth Biannual CASP (Critical Assessment of Techniques for Protein Structure Prediction) experiment. The goal is to utilize these approaches together with the immense computer power that can be harnessed through the internet and volunteers all over the world (you!) to address critical biomedical questions of protein-related diseases. Predictor@home is a pilot project of the Berkeley Open Infrastructure for Network Computing (BOINC)

Join Predictor
- Rules and policies **[read this first]**
- Getting started
- Create account

Returning participants
- Log in
- User page - view stats, modify preferences
- Teams - create or join a team
- Download Predictor/BOINC client

Community
- User profiles
- Message boards
- Questions and problems

Project totals and leader boards
- Top users
- Top hosts
- Top teams

Science
- Current Progress
- Learn About Predictor
- Science Links

About Predictor
- Project Team and Plan
- Photo Gallery

Predictor Statistics
- Current Total Statistics
- Individual User Statistics
- Group Statistics
- Country Statistics

News

5/24/04 - If you need some more help to install your BOINC client, just go here. :-)

4/20/04 - We should have predictor up and running on its final site this week. Need to do some testing of the web server security.

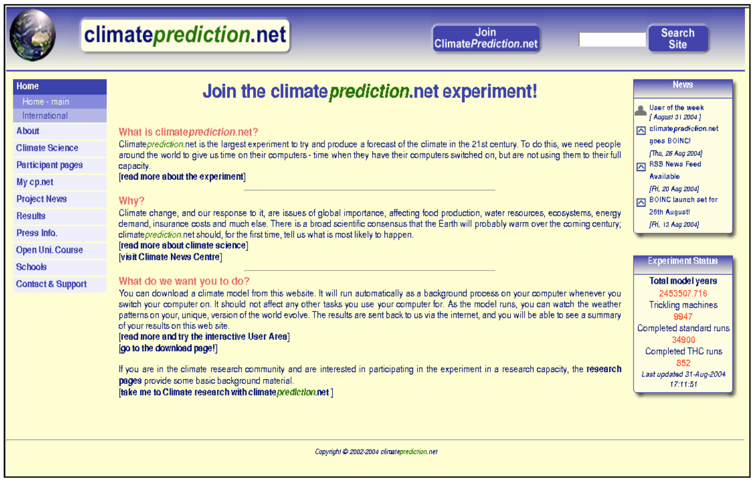
3/16/04 - At last we have news, and it's good. Michela is working hard to complete this project and Chris is just playing with the webpage.

Current Targets

Target 1
Target 2
Target 3
Target 4
Target 5

Supported by the NIH
Center for Multiscale Modeling Tools in Molecular Biology (MMTSB)
and the NSF
Center for Theoretical Biological Physics (CTBP)

Climateprediction.net



Join the climateprediction.net experiment!

What is climateprediction.net?

Climateprediction.net is the largest experiment to try and produce a forecast of the climate in the 21st century. To do this, we need people around the world to give us time on their computers - time when they have their computers switched on, but are not using them to their full capacity.
[**read more about the experiment**]

Why?

Climate change, and our response to it, are issues of global importance, affecting food production, water resources, ecosystems, energy demand, insurance costs and much else. There is a broad scientific consensus that the Earth will probably warm over the coming century; climateprediction.net should, for the first time, tell us what is most likely to happen.
[**read more about climate science**]
[**visit Climate News Centre**]

What do we want you to do?

You can download a climate model from this website. It will run automatically as a background process on your computer whenever you switch your computer on. It should not affect any other tasks you use your computer for. As the model runs, you can watch the weather patterns on your, unique, version of the world evolve. The results are sent back to us via the internet, and you will be able to see a summary of your results on this web site.
[**read more and try the interactive User Area**]
[**go to the download page!**]

If you are in the climate research community and are interested in participating in the experiment in a research capacity, the **research pages** provide some basic background material.
[**take me to Climate research with climateprediction.net**]

News

User of the week
[August 31 2004]

climateprediction.net goes BOINC!
[Thu, 26 Aug 2004]

RSS News Feed Available
[Fri, 20 Aug 2004]

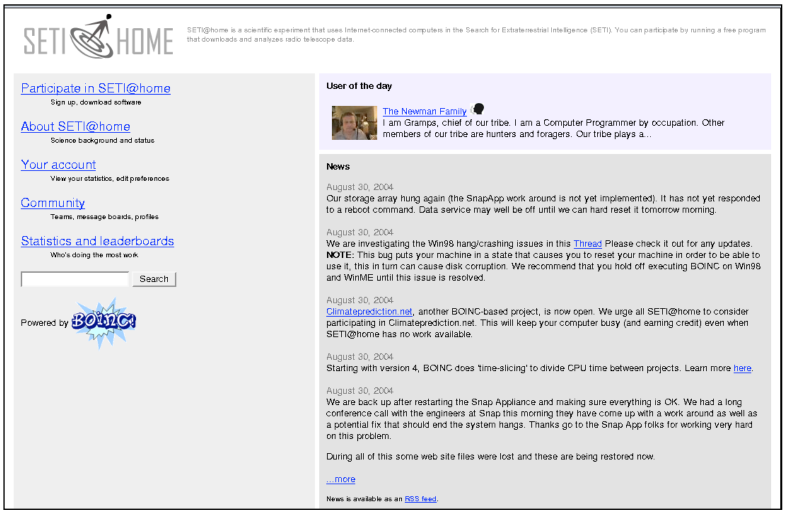
BOINC launch set for 26th August!
[Fri, 13 Aug 2004]

Experiment Status

Total model years
2453507.716
Trickling machines
9947
Completed standard runs
34900
Completed THC runs
852
Last updated 31-Aug-2004 17:11:51

Copyright © 2002-2004 climateprediction.net

SETI@home


SETI HOME
Participate in SETI@home
Sign up, download software
About SETI@home
Science background and status
Your account
View your statistics, edit preferences
Community
Teams, message boards, profiles
Statistics and leaderboards
Who's doing the most work
Search
Powered by BOINC!

SETI@home is a scientific experiment that uses Internet-connected computers in the Search for Extraterrestrial Intelligence (SETI). You can participate by running a free program that downloads and analyzes radio telescope data.

User of the day

The Newman Family
I am Gramps, chief of our tribe. I am a Computer Programmer by occupation. Other members of our tribe are hunters and foragers. Our tribe plays a...

News

August 30, 2004
Our storage array hung again (the SnapApp work around is not yet implemented). It has not yet responded to a reboot command. Data service may well be off until we can hard reset it tomorrow morning.

August 30, 2004
We are investigating the Win98 hang/crashing issues in this Thread Please check it out for any updates. **NOTE**: This bug puts your machine in a state that causes you to reset your machine in order to be able to use it, this in turn can cause disk corruption. We recommend that you hold off executing BOINC on Win98 and WinME until this issue is resolved.

August 30, 2004
Climateprediction.net, another BOINC-based project, is now open. We urge all SETI@home to consider participating in Climateprediction.net. This will keep your computer busy (and earning credit) even when SETI@home has no work available.

August 30, 2004
Starting with version 4, BOINC does 'time-slicing' to divide CPU time between projects. Learn more here.

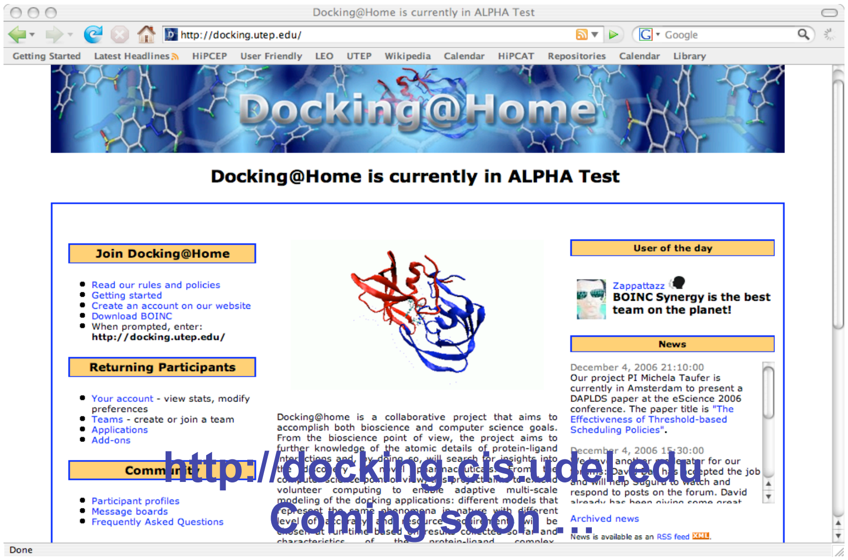
August 30, 2004
We are back up after restarting the Snap Appliance and making sure everything is OK. We had a long conference call with the engineers at Snap this morning they have come up with a work around as well as a potential fix that should end the system hangs. Thanks go to the Snap App folks for working very hard on this problem.

During all of this some web site files were lost and these are being restored now.

...more

News is available as an RSS feed.

Docking@Home: DAPLDS Portal



Docking@Home is currently in ALPHA Test

Join Docking@Home

- Read our rules and policies
- Getting started
- Create an account on our website
- Download BOINC
- When prompted, enter: **http://docking.utep.edu/**

Returning Participants

- Your account - view stats, modify preferences
- Teams - create or join a team
- Applications
- Add-ons

Community

- Participant profiles
- Message boards
- Frequently Asked Questions

Docking@home is a collaborative project that aims to accomplish both bioscience and computer science goals. From the bioscience point of view, the project aims to further knowledge of the atomic details of protein-ligand interactions and, by doing so, will search for insights into the discovery of novel pharmaceuticals. From the computer science point of view, this project aims to extend volunteer computing to enable adaptive multi-scale modeling of the docking applications: different models that represent the same phenomena in nature with different level of accuracy and resource requirements will be chosen at run time based on results collected so far and characteristics of the protein-ligand complex.

User of the day

Zappattazz
BOINC Synergy is the best team on the planet!

News

December 4, 2006 21:10:00
Our project PI Michela Taufer is currently in Amsterdam to present a DAPLDS paper at the eScience 2006 conference. The paper title is "The Effectiveness of Threshold-based Scheduling Policies".

December 4, 2006 15:30:00
We have another moderator for our forums: David [illegible] has accepted the job and will help Suguru to watch and respond to posts on the forum. David already has been giving some great

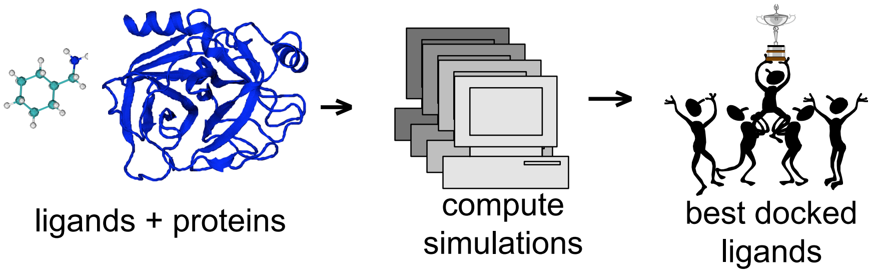
Archived news

News is available as an RSS feed XML.

http://docking.cis.udel.edu

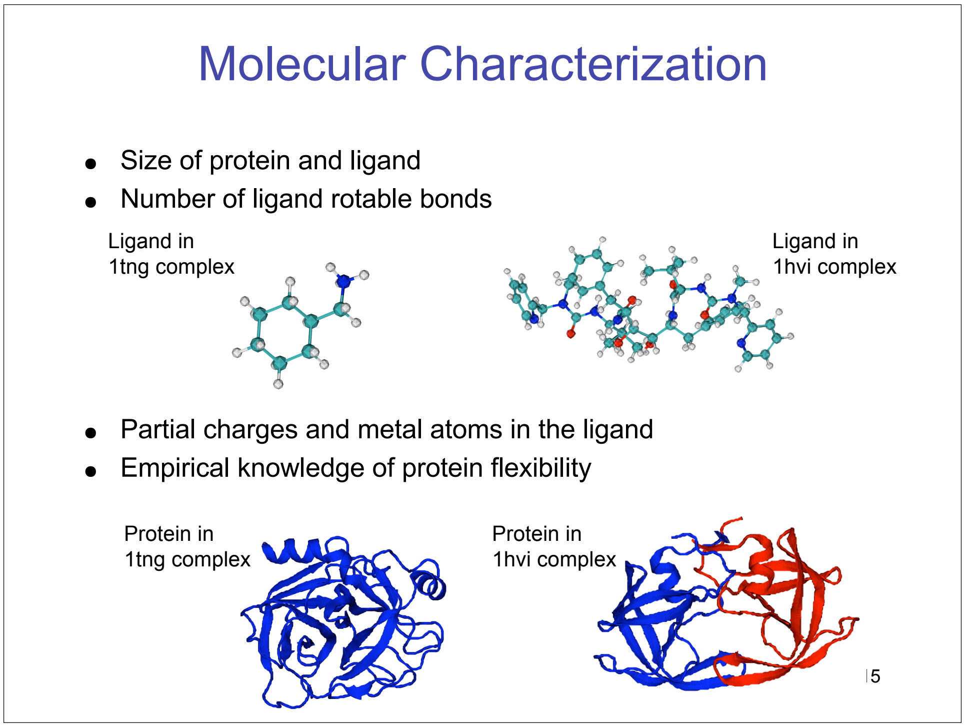
Coming soon…

DAPLDS Overview



ligands + proteins → compute simulations → best docked ligands

- Search for algorithms that provide accurate results
- Search for an computers that can execute these algorithms



Molecular Characterization

- Size of protein and ligand
- Number of ligand rotable bonds

Ligand in
1tng complex

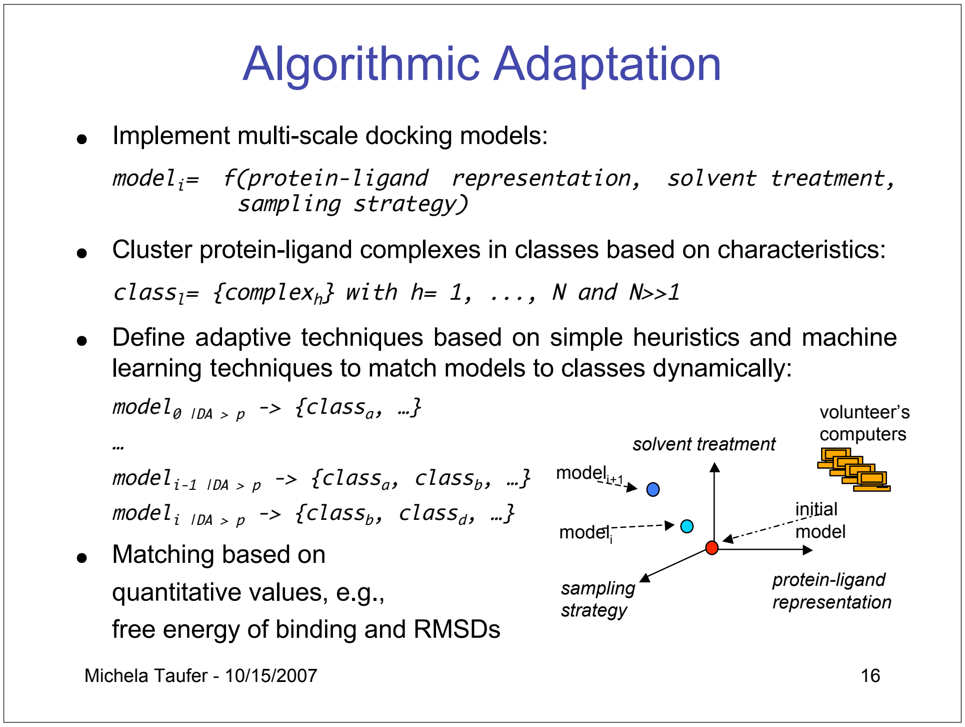
Ligand in
1hvi complex

- Partial charges and metal atoms in the ligand
- Empirical knowledge of protein flexibility

Protein in
1tng complex

Protein in
1hvi complex

15



Algorithmic Adaptation

- Implement multi-scale docking models:

  $model_i$= f(protein-ligand representation, solvent treatment, sampling strategy)

- Cluster protein-ligand complexes in classes based on characteristics:

  $class_l$= {$complex_h$} with h= 1, ..., N and N>>1

- Define adaptive techniques based on simple heuristics and machine learning techniques to match models to classes dynamically:

  $model_{0\ \mid DA\ >\ p}$ -> {$class_a$, …}

  …

  $model_{i-1\ \mid DA\ >\ p}$ -> {$class_a$, $class_b$, …}

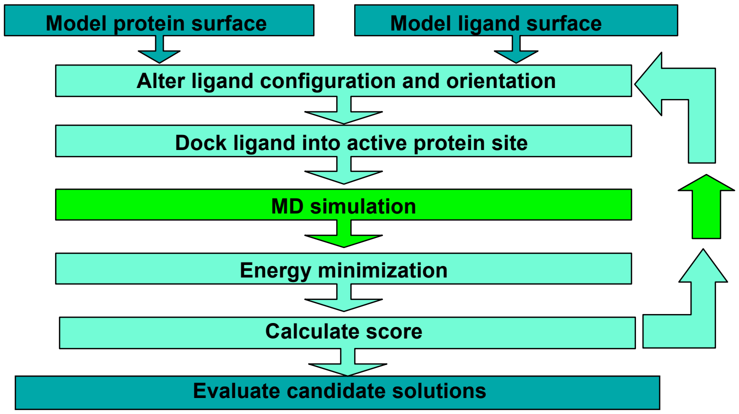
  $model_{i\ \mid DA\ >\ p}$ -> {$class_b$, $class_d$, …}

- Matching based on quantitative values, e.g., free energy of binding and RMSDs

Michela Taufer - 10/15/2007

16

Model

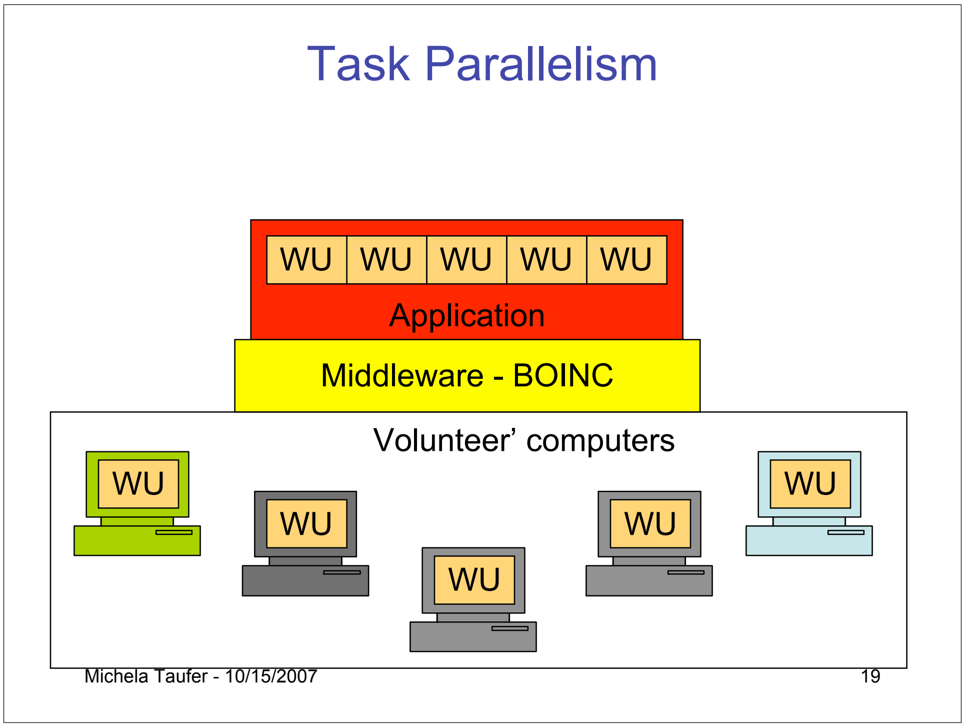

Model protein surface
Model ligand surface
Alter ligand configuration and orientation
Dock ligand into active protein site
MD simulation
Energy minimization
Calculate score
Evaluate candidate solutions

Task Parallelism in VC

- VC projects search for computational solutions that are close to results in Nature
- Searches are partitioned into large numbers of work-units (*WU*s) and implemented via large-scale simulations:

$$\mathtt{simulation} \equiv \left\langle WU_j \middle| j = 1,\ldots,N \right\rangle$$

- *WU$_j$* are characterized by a computational model and initial conditions
- Scientists can quantify quality of *WU* results
  - Quantitative measurement of binding affinity: e.g., free energy
- *WU*s are replicated to assure validity of final results
  - *WU* replicas assigned to different volunteer computers
  - Sanity check is performed on the replicas
  - New replicas are automatically generated if errors or time-outs occur



Task Parallelism

WU WU WU WU WU

Application

Middleware - BOINC

Volunteer' computers

WU

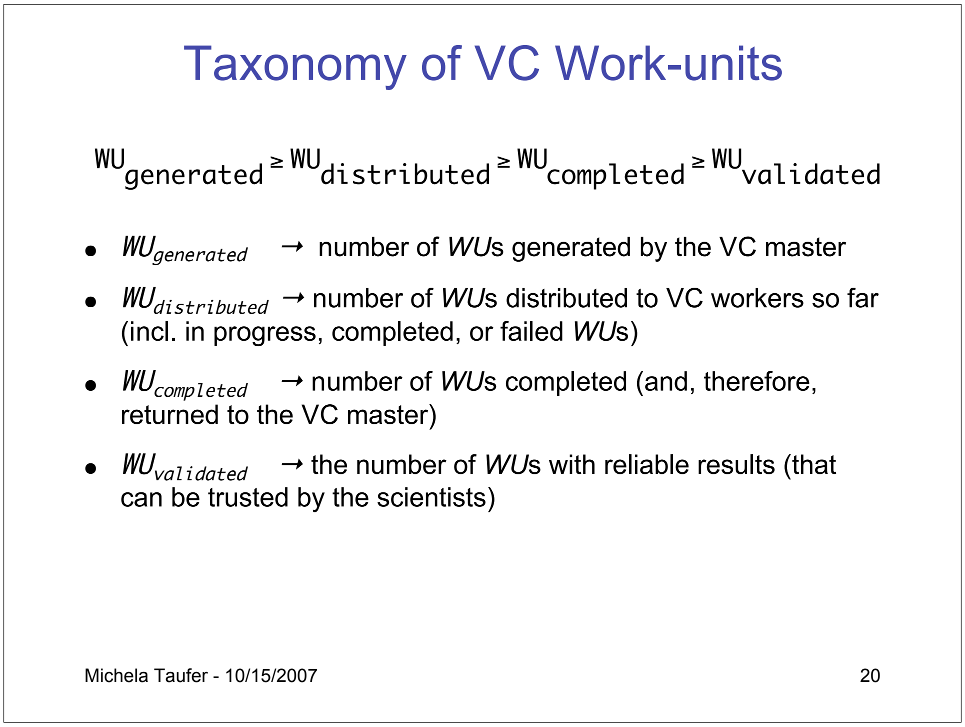
WU

WU

WU

WU

Michela Taufer - 10/15/2007 19

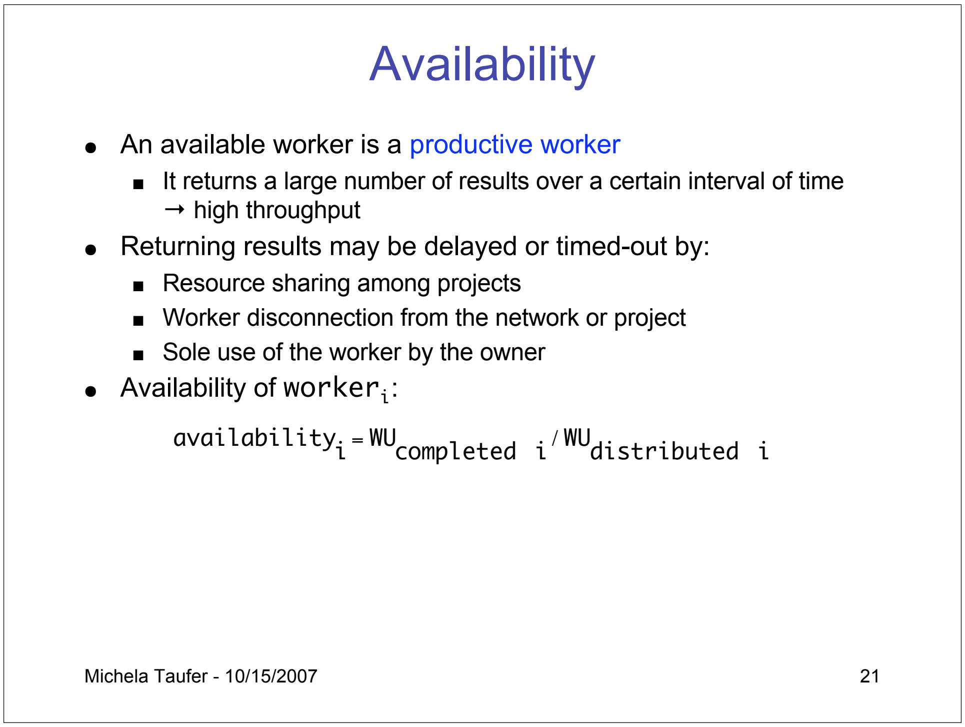


Taxonomy of VC Work-units

$$\mathtt{WU}_{\mathtt{generated}} \geq \mathtt{WU}_{\mathtt{distributed}} \geq \mathtt{WU}_{\mathtt{completed}} \geq \mathtt{WU}_{\mathtt{validated}}$$

- $WU_{generated}$ → number of *WU*s generated by the VC master
- $WU_{distributed}$ → number of *WU*s distributed to VC workers so far (incl. in progress, completed, or failed *WU*s)
- $WU_{completed}$ → number of *WU*s completed (and, therefore, returned to the VC master)
- $WU_{validated}$ → the number of *WU*s with reliable results (that can be trusted by the scientists)

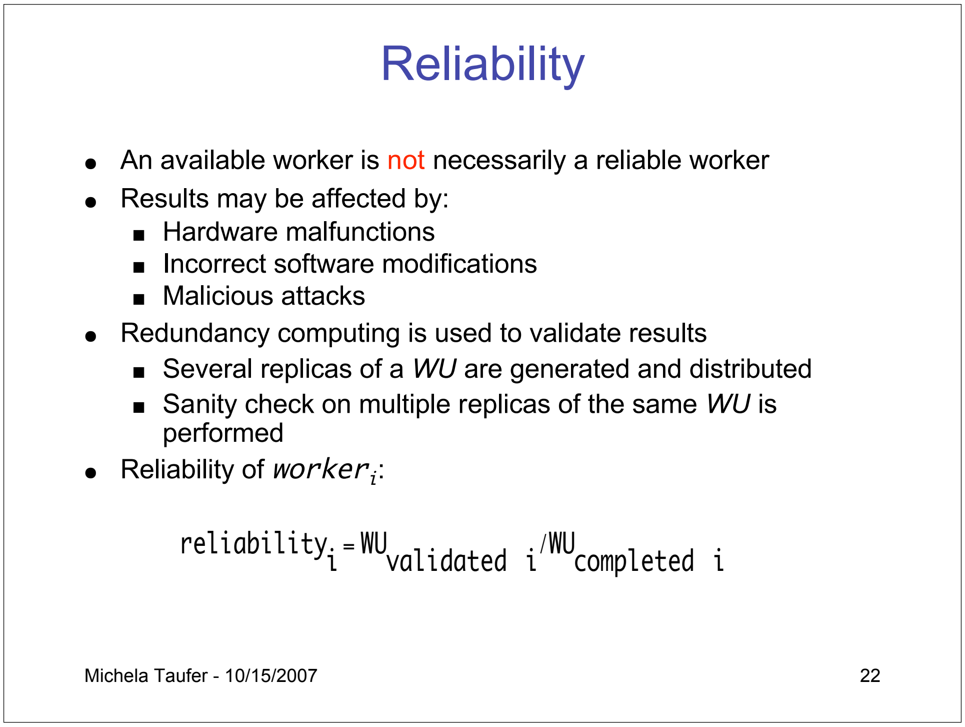
Michela Taufer - 10/15/2007 20



Availability

- An available worker is a productive worker
  - It returns a large number of results over a certain interval of time → high throughput
- Returning results may be delayed or timed-out by:
  - Resource sharing among projects
  - Worker disconnection from the network or project
  - Sole use of the worker by the owner
- Availability of $\mathtt{worker}_i$:

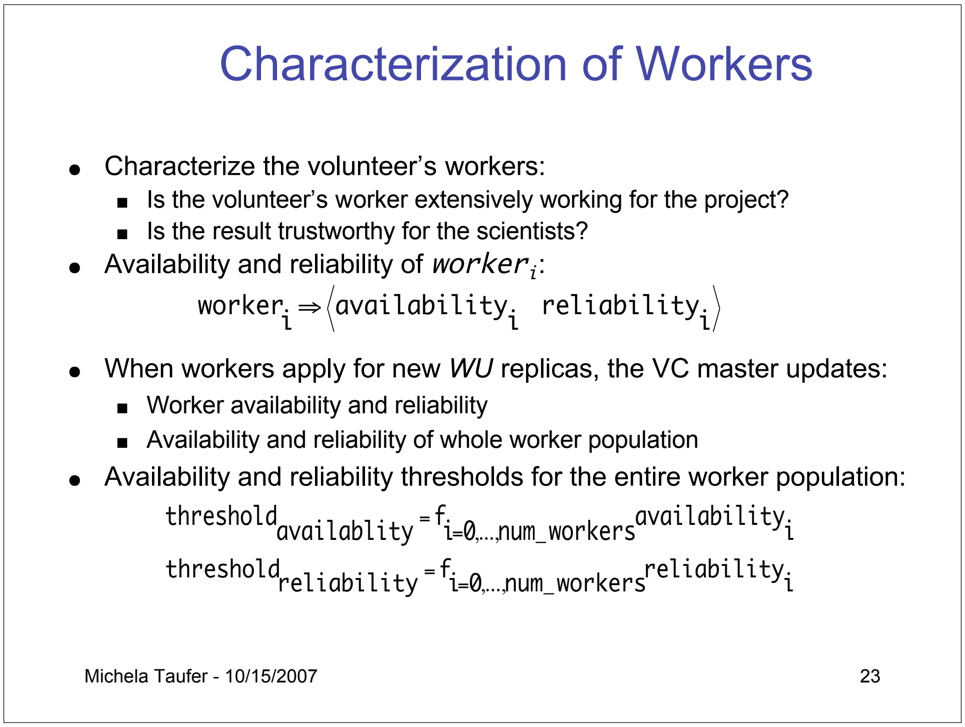
$$\mathtt{availability}_i = WU_{\mathtt{completed\ i}} / WU_{\mathtt{distributed\ i}}$$



Reliability

- An available worker is not necessarily a reliable worker
- Results may be affected by:
  - Hardware malfunctions
  - Incorrect software modifications
  - Malicious attacks
- Redundancy computing is used to validate results
  - Several replicas of a *WU* are generated and distributed
  - Sanity check on multiple replicas of the same *WU* is performed
- Reliability of $worker_i$:

$$\mathtt{reliability}_i = WU_{\mathtt{validated\ i}} / WU_{\mathtt{completed\ i}}$$



Characterization of Workers

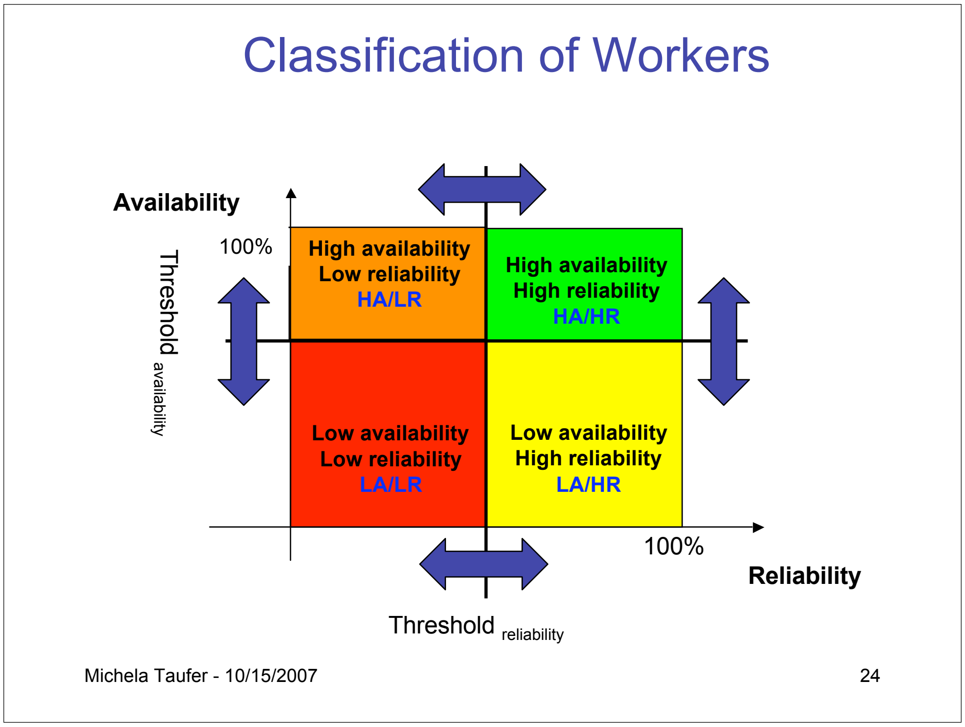
- Characterize the volunteer's workers:
  - Is the volunteer's worker extensively working for the project?
  - Is the result trustworthy for the scientists?
- Availability and reliability of *worker*$_i$:

$$\mathtt{worker}_i \Rightarrow \langle \mathtt{availability}_i \;\; \mathtt{reliability}_i \rangle$$

- When workers apply for new *WU* replicas, the VC master updates:
  - Worker availability and reliability
  - Availability and reliability of whole worker population
- Availability and reliability thresholds for the entire worker population:

$$\mathtt{threshold}_{\mathtt{availablity}} = f_{\mathtt{i}=0,\ldots,\mathtt{num\_workers}} \mathtt{availability}_{\mathtt{i}}$$

$$\mathtt{threshold}_{\mathtt{reliability}} = f_{\mathtt{i}=0,\ldots,\mathtt{num\_workers}} \mathtt{reliability}_{\mathtt{i}}$$



Classification of Workers

Scheduling Policies

- First-come first-serve:
  - Replicas are assigned to any worker - similar to having availability and reliability thresholds both equal to 0
- Fixed thresholds:
  - Replicas are assigned to workers that have availability and reliability above fixed thresholds
- Variable thresholds:
  - The assignment of replicas reflects runtime changes in the volunteer community
- Compound of simple heuristics driven by genetic algorithms
  - Population of chromosomes → conditional rules for work-unit assignments
  - Mutation and crossover on parameters and metrics that characterize workers in chromosomes
- Homogenous redundancy
  - Send replicas to numerically equivalent workers, demand identical answers (*M. Taufer et al., HWC'05*)

Evaluation

- Challenges in testing new policies on VC projects:
  - Designers cannot predict the effect of their decisions
  - Time to measure and compare performance would be too long
  - Problems due to testing might upset volunteers
  - Every experiment is unique and unrepeatable
- Testing on a reduced-size system is not indicative of real VC projects
- Need for a simulation environment that allows project designers to tailor the simulation to the volunteer computing community
  - Simulator of BOINC Applications or SimBA

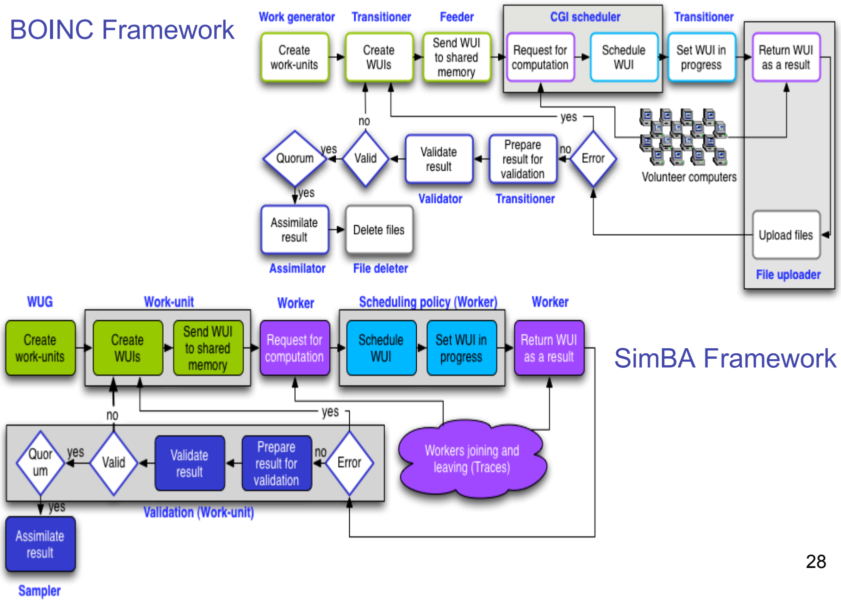
SimBA

- SimBA (Simulator of BOINC Applications) is a discrete event simulator written in Python, that models the behavior of a BOINC master (*T. Estrada et al., eScience'06*)

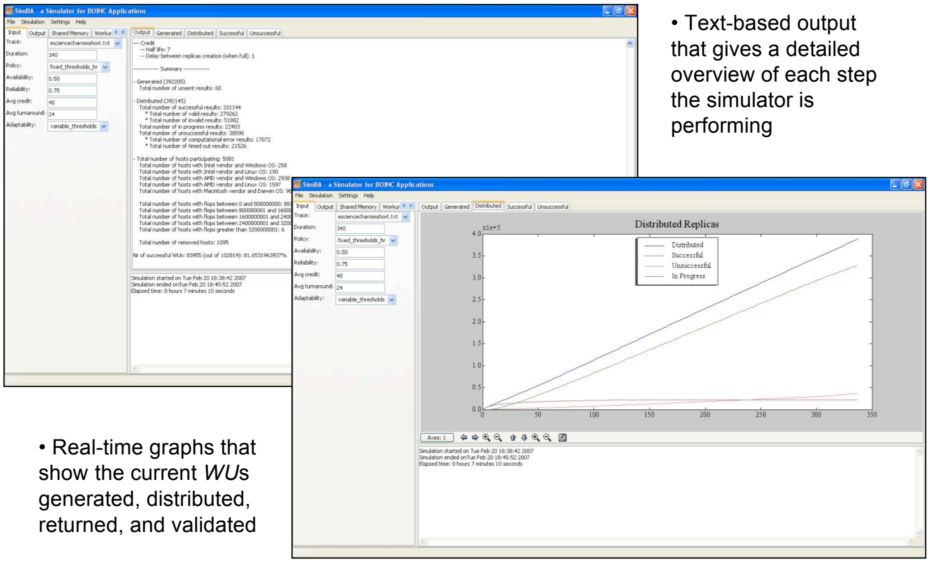
- Entities are:
  - work-unit generator
  - worker generator
  - worker
  - work-unit
  - work-unit replica
  - sampler (monitor)

- Events are:
  - generate work-unit
  - generate worker
  - request instances
  - generate instances
  - determine instance output
  - check if simulation is over

BOINC vs. SimBA



SimBA User Interface



SimBA - a Simulator for BOINC Applications

Trace: esciencecharmmshort.txt
Duration: 340
Policy: fixed_thresholds_hr
Availability: 0.50
Reliability: 0.75
Avg credit: 40
Avg turnaround: 24
Adaptability: variable_thresholds

```
--- Credit
  -- Half life: 7
  -- Delay between replicas creation (when full): 1

------------- Summary -------------

- Generated (392205)
   Total number of unsent results: 60

- Distributed (392145)
   Total number of successful results: 331144
     * Total number of valid results: 279262
     * Total number of invalid results: 51882
   Total number of in progress results: 22403
   Total number of unsuccessful results: 38598
     * Total number of computational error results: 17072
     * Total number of timed out results: 21526

- Total number of hosts participating: 5081
   Total number of hosts with Intel vendor and Windows OS: 258
   Total number of hosts with Intel vendor and Linux OS: 190
   Total number of hosts with AMD vendor and Windows OS: 2938
   Total number of hosts with AMD vendor and Linux OS: 1597
   Total number of hosts with Macintosh vendor and Darwin OS: 98

   Total number of hosts with flops between 0 and 800000000: 881
   Total number of hosts with flops between 800000001 and 16000
   Total number of hosts with flops between 1600000001 and 2400
   Total number of hosts with flops between 2400000001 and 3200
   Total number of hosts with flops greater than 3200000001: 6

   Total number of removed hosts: 1095

Nr of successful WUs: 83955 (out of 102819): 81.6531963937%
```

```
Simulation started on Tue Feb 20 18:38:42 2007
Simulation ended onTue Feb 20 18:45:52 2007
Elapsed time: 0 hours 7 minutes 10 seconds
```

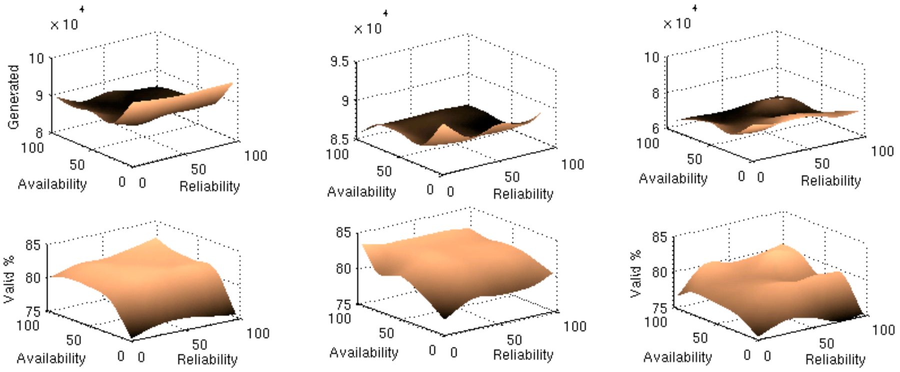
- Text-based output that gives a detailed overview of each step the simulator is performing

- Real-time graphs that show the current *WU*s generated, distributed, returned, and validated

Samples of SimBA Outputs

P@h
Monte Carlo application

World Community Grid - IBM
F@h **Defeat Cancer**



A Case Study: Description

- Traces from an existing VC project: P@h
- Minimum number of replicas per *WU*: 3
- Number of workers in P@h traces: 14,000
- Duration of VC project: 12 days
- Scheduling policies:
  - First-Come First-Serve
  - Fixed thresholds - 75% availability and 75% reliability
  - Variable thresholds - availability and reliability change at runtime based on number of pending *WU*s

Our Question: What scheduling policy does provide us with a better throughput and resource utilization over 12 days of VC project?

A Case Study: Results

P@h performance of different scheduling polices based on SimBA results

P@h	FCFS	Fixed Thresholds (75% - 75%)		Variable Thresholds	
Generated WU	78,658	78,886	+0.2%	78621	~0%
Generated WU replicas	284,140	252,258	**-11.2%**	253,611	**-10.7%**
Error replicas	38,491	13,802	**-64.1%**	14,424	**-62.5%**
Valid WU (throughput)	70,948	71,201	**+0.4%**	72,929	**+2.8%**
Avg. replicas per WU	3.6	3.2	**-11.1%**	3.2	**-11.1%**

- Dynamically adaptable thresholds keep track of runtime changes in the VC community (*M. Taufer et al., PCGrid'07*)
  - Average replicas per *WU* drops from 3.6 to 3.2 for thresholds based policies
  - Previously ineligible volunteer computers are regained for the project

Conclusions

- Volunteer computing is a powerful paradigm for applications that use task parallelism
- Interesting research topics in volunteer computing include:
  - Improve accuracy of models for the simulations of protein-ligand interactions
  - Study of effective scheduling polices for the selection of volunteer’s computers
- Several applications in science can benefit from effectives volunteer computer systems
  - Geology
  - Biology
  - Climate studies
- Computer scientists working in this field are exposed to interdisciplinary research
- If you want to know more, contact me at taufer@udel.edu or come by my office at Smith Hall 406