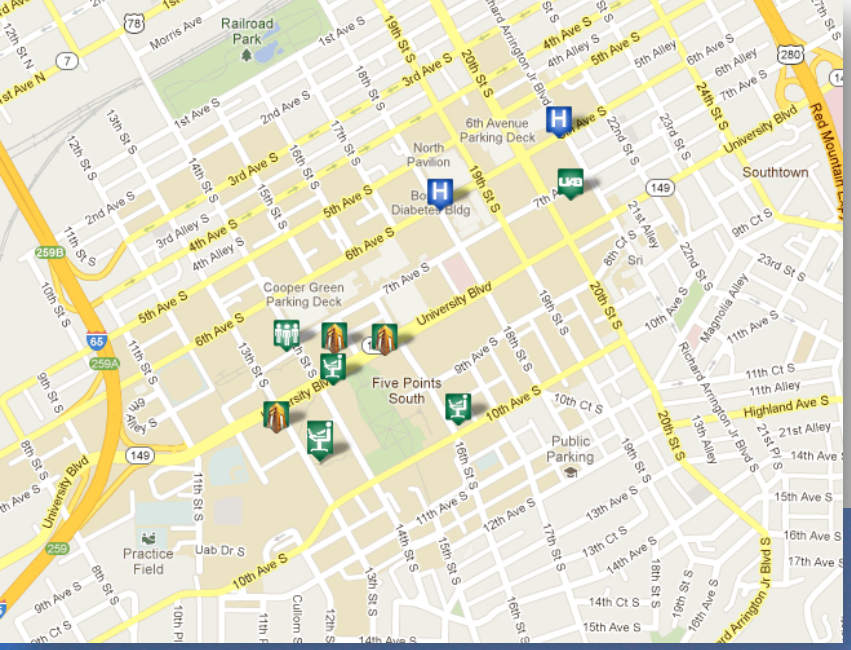
Condor @ UAB

Expanding the Campus Computing Cloud

Condor Week 2012

John-Paul Robinson jpr@uab.edu
Poornima Pochana ppreddy@uab.edu
Thomas Anthony tanthony@uab.edu

UAB INFORMATION TECHNOLOGY

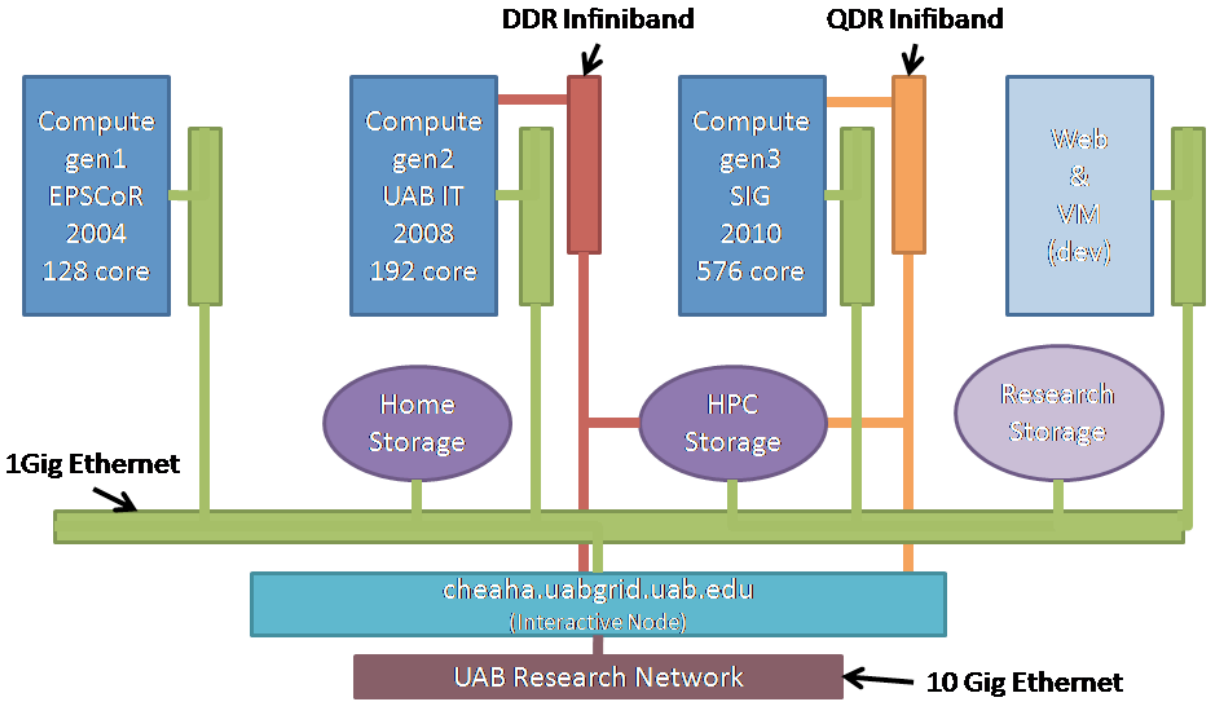


THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

- Public University with 17,571 Student Body and 11,240 Faculty and Staff*
- Ranked 20th in NIH Funding, 32nd in Federal
- Large Academic Medical Center
- Established in 1969

UAB Research Cloud

- Operated by UAB IT Research Computing Group
- 868 cores, three hardware generations, one SGE scheduler
- ~9Tflops compute power, 180TB High performance storage
- Research Storage (40TB and growing)
- Virtual Machine fabric with OpenNebula



https://docs.uabgrid.uab.edu/wiki/Cheaha

UAB Condor Pilot

Can idle desktops help support computationally intense research at UAB?

– Dr. Charlie Prince, Assistant VP for Research

- Project Partners
  - UAB IT Research Computing
  - Department of Computer and Information Science
  - UAB IT Desktop Computing
- Science Domain
  - Structural Biology – a Molecular Docking workflow

Molecular Docking

- Molecular docking is a method to find the best match between two molecules: a receptor and a ligand
  - representation of the system
  - conformational space search
  - ranking of potential solutions
- Virtual Screening
  - high performance computing
  - large databases of chemical compounds
  - identify possible drug candidates
- Active research area at UAB
- Molecular Docking in Context
  - big for computing but small part of x-ray crystallography workflow
  - http://vimeo.com/7643687 -- Claude Sauter

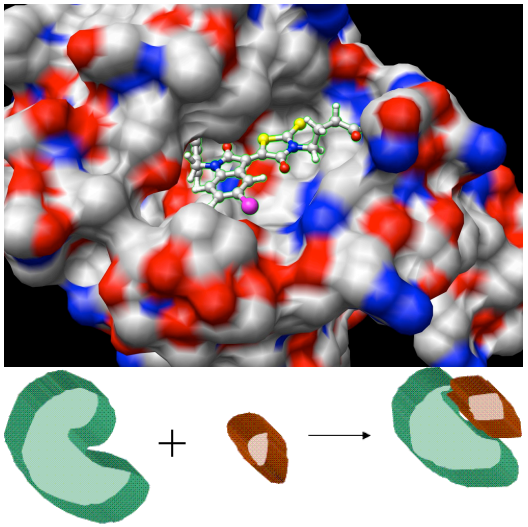
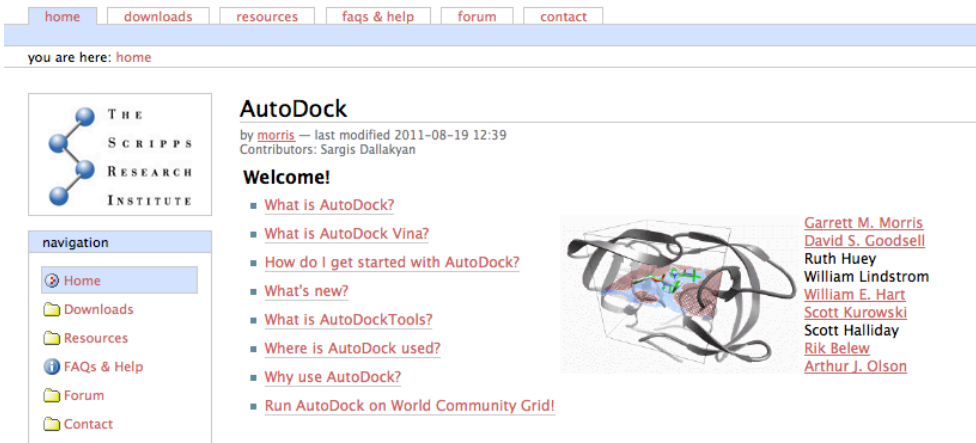


Image credit: Wikipedia

Autodock

- Automated docking software for predicting optimal ligand-protein interaction
- Uses heuristic search to minimize energy using Lamarkian genetic algorithms or other search like simulated annealing
- GPL Software License

AutoDock http://autodock.scripps.edu



High-Throughput Docking Workflow

- Virtual Screening with Autodock
  - Prepare ligand-receptor coordinate files
  - Calculate affinity grid for the receptor molecule
  - Docking simulation
- Docking Computing Requirements
  - Directly proportional to number of ligands, number of receptors and complexity and flexibility of molecules
  - Our example workflow contains 1088 jobs representing 5436 ligand-receptor dockings

Condor

- Features
  - Workload management system for compute-intensive jobs that can work independently – "Pleasantly Parallel"
  - Developed for "cycle stealing" from idle machines, good for dedicated computing too
  - Supported on a large number of systems
  - Get started easily with personal instances
  - Good for cloud computing, can set up condor pool on Amazon EC2
- Community
  - Active and engaged developer community
  - Broad adoption in many domains
  - Freely available under Apache license

Condor
High Throughput Computing

http://research.cs.wisc.edu/condor/

Pilot Steps

- Create pilot campus Condor pool
- Understand original molecular docking workflow for SGE HPC cluster
- Convert existing job to work on Condor job scheduler
- Analyze performance on different fabrics
  - UAB Campus Compute Cluster – 878 core, production (our control)
  - UAB Pilot Condor Pool – 60 core, pilot system
  - U. Wisconsin CHTC Condor Pool – 1000+ core, production
  - OSG Engage VO via Condor-G and glide-in WMS – 80-100k core, production

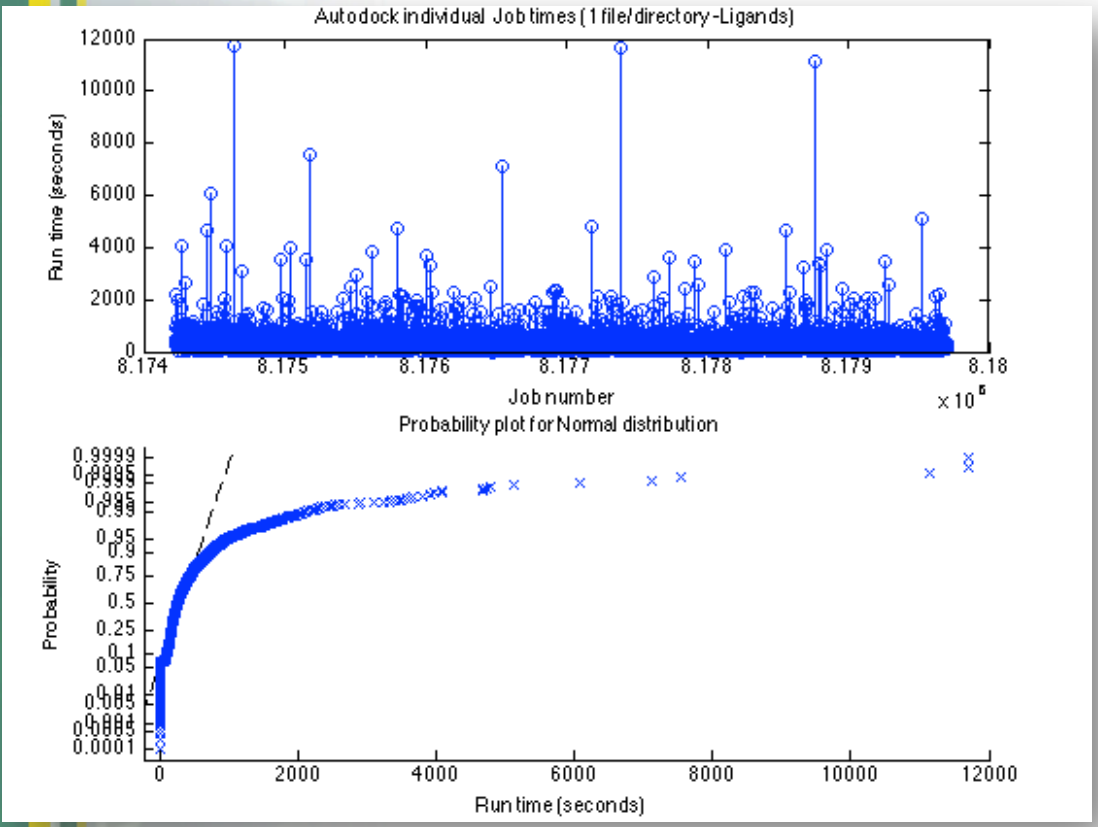
Pilot Condor Pool at UAB

- 20 64-bit Linux workstations and Condor master provided by CIS
- 40 32-bit Windows desktops from IT Desktop Services
- Workflow migration, fabric analysis and support by IT Research Computing

Performance and Job Structure

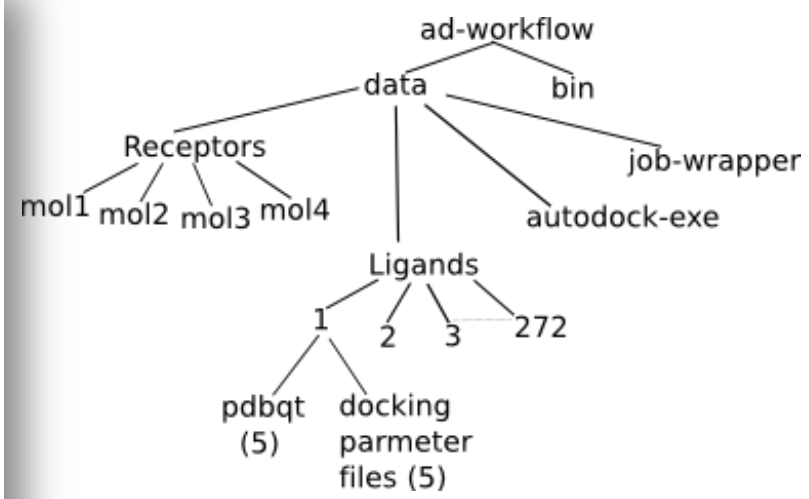
Run-time Distribution per Docking

- 90% of dockings complete in 15 minutes or less



Static Docking Structure Workflow

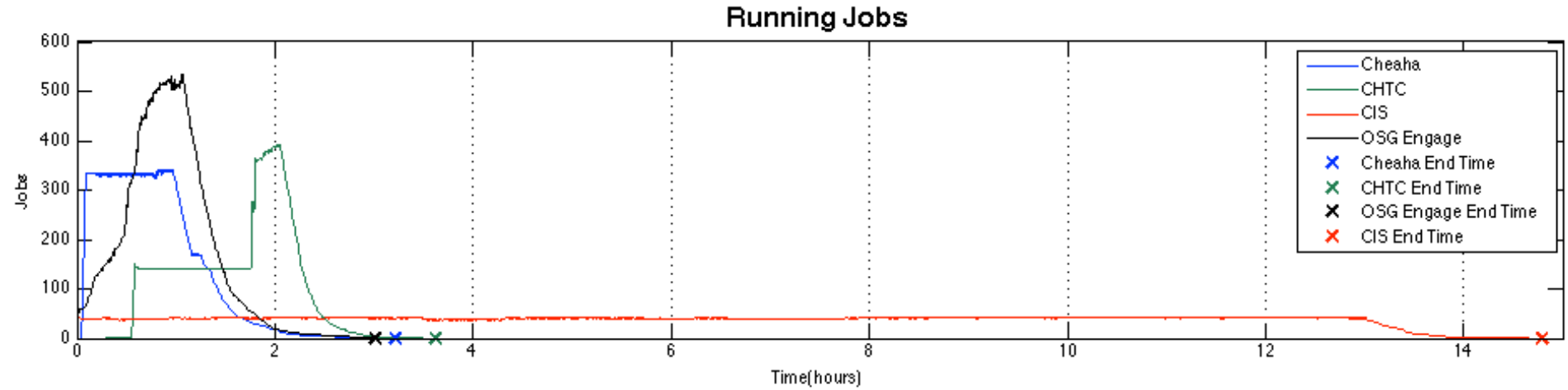
- 5 ligands per 1 receptor
- Identical structure for all fabrics

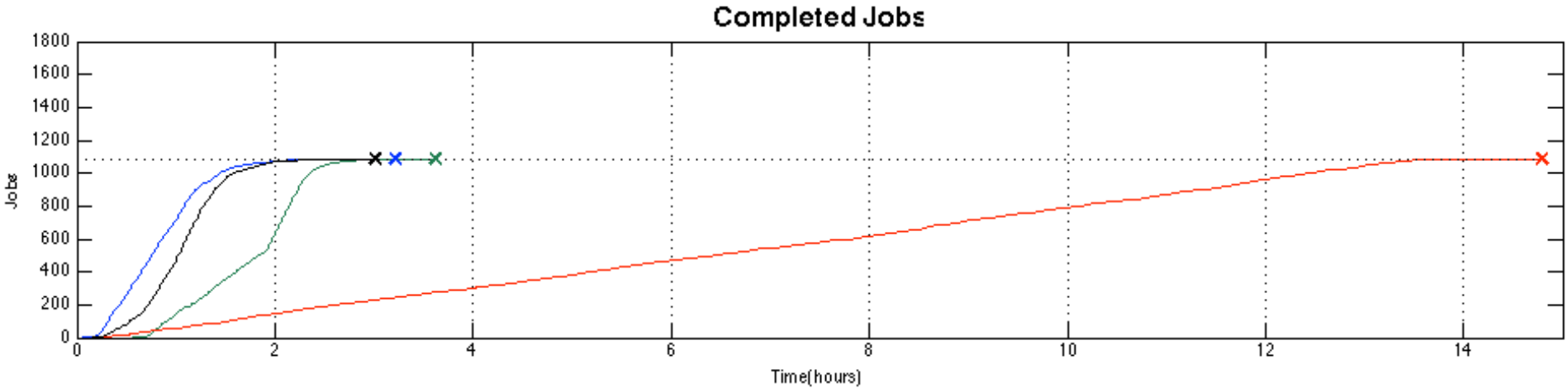


Computing Performance

Summary:

- 1088 compute jobs containing 5 ligands and 1 receptor each
- Each fabric varies in available compute cores and application scales linearly in response
- Campus Condor pool @ CHTC and spare cycles on OSG perform as well as UAB's dedicated compute cluster





Conclusions

- Applications
  - Condor resources perform as well as campus cluster for high-throughput applications
  - High-throughput workflows can scale linearly
  - Dynamic workload scheduling could balance jobs
- Hardware
  - Linux workstations better match scientific computing
  - Cygwin makes Windows more useful for scientific
  - 64-bit platforms much better than 32-bit platforms
- Usability
  - Condor makes workflow migration between campus cluster, campus workstations, and national grids easy

Future Work

- Recommend implementation of Condor @ UAB
- Recommend high-throughput workflows adopt Condor instead of SGE
- Leverage spare cycle computing in other areas of infrastructure
- Follow our progress http://projects.uabgrid.uab.edu/condor

Special Thanks

- CHTC
  - Miron Livny, Brooklin Gore, and Bill Taylor for Condor guidance and access to CHTC Condor pool @ University of Wisconsin
- OSG Engage
  - John McGee and Mats Rynge for access to and support of Engage VO resources on the Open Science Grid
- UAB
  - Larry Owen in UAB Computer and Information Sciences Department, Sterling Griffin and Susan Wells in UAB IT Desktop Services
  - Doug Rigney, Interim VP UAB IT, for on-going support of research computing activities

Additional Autodock Performance Analyses in the Literature

- Ellingson S. et. al. *High-Througput Virtual Molecular Docking: Hadoop Implementation of Autdock4 on a Private Cloud (2011)*
- Norgan A., et. al. *Multilevel Parallelization of Autodock 4.2 (2011)*
- Collingnon B., et. al. *Task-Parallel Message Passing Interface Implementation of Autodock4 for Docking of Very Large Databases of Compounds Using High-Performance Super-Computers (2010)*
- Huang N., et. al. *Benchmarking Sets for Molecular Docking (2006)*