October 18, 2013

Configuration and Deployment of a Virtual Cluster for Molecular Docking Experiments on the PRAGMA Cloud

Karen Rodriguez, Kevin Lam, Dr. Jason Haga, Dr. Kohei Ichikawa

Department of Bioengineering, University of California San Diego, La Jolla, CA 92093, USA
Software Design and Analysis Laboratory, Nara Institute of Science and Technology, Ikoma, Nara Prefecture 630-0192, Japan

Abstract

Molecular docking simulations are an important tool that is often employed in biomedical research and pharmaceutical industries to scan compound databases in pursuit of a specific molecular interaction. This is often used in drug screening experiments when an entire drug compound database cannot realistically be tested with assays on a wet laboratory bench. DOCK is a computer application developed to aid researchers by simulating molecular bindings between a single cellular receptor and a list of ligands. DOCK then calculates an estimate of the stability of the bond resulting from a number of conformations for each ligand. This allows one to obtain a reduced list of compounds that are more likely to interact with the receptor of interest in a desired way in real life. The result would narrow down experiments to only include compounds expected to bind to a receptor in a certain way; thus allowing scientists to finish such experiments within a reasonable time and economic cost. DOCK, however, is a demanding program in terms of processing power. Additionally, this is known to not yield consistent results when executed from machines with varying processors, operating systems, and compilers. Availability of a reliable environment on which DOCK can be executed and yield reliable and quick results is therefore very important for large-scale experiments of drug screening nature. The purpose of this project was to create such environment out of networked virtual machine (VM) clones and upload it onto the PRAGMA cloud in order to meet DOCK's high computational power demands whilst keeping computation times

reasonable and results accurate. Ultimately, a 64bit virtual machine with CentOS 5.9 (32bit), gcc compiler version 4.1.2, DOCK 6.2 and MPICH was successfully built, cloned, networked, and deployed onto both the National Institute for Advanced Industrial Science and Technology and University of California, San Diego hosts; from which it operates with the desired features thus providing the needed environment for a large-scale docking workflow.

Introduction

DuSPs (Dual-Specificity Phosphatases) are mammalian enzymes whose activation/inhibition regulates the production of actin filaments; which in turn affects several cellular processes in several cell types throughout the human body. Finding compounds that can control their inhibition is thus of great interest to researchers. Virtualized screenings of docked compounds are an important tool in this interest, as they spare time and efforts to researchers interested in DuSP-specific inhibition factors. This task can be completed utilizing DOCK, a software suite developed by the University of California, San Francisco. DOCK simulates molecular binding of a ligand and cellular receptor. As it computes a set input number of conformations, it evaluates bond strength and assigns a score to each instance. Lower scores reflect stronger bonds, and thus those most stable and likely to occur in a real biological setting. If a list of ligands is input instead of a single compound, DOCK will scan each compound against a single receptor and output the respective scores. This tool is often employed in the field of biotechnology and pharmaceuticals to narrow down large drug compound databases (such as ZINC) to a smaller and more manageable subset of molecules that will elicit a

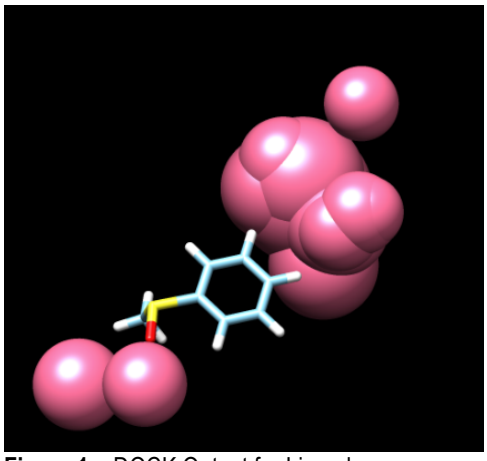

Figure 1 – DOCK Output for Ligand ZINC0013564 Against the Test Suite Receptor.

desired interaction. Results can then be verified on the lab bench, thus saving time by not carrying out these experiments for millions of ligands. Thus, it is important to establish a reliable computational environment that can carry out these experiments within a realistic amount of time. DOCK, however, is a computationally intensive application and therefore requires some time for each computation. For large ligand lists, the computation times will add up and could take weeks, months, or years depending on the size of the screening inputs. Grid computing was previously investigated as an alternative by networking a series of physical nodes such that a messenger-passing application could distribute jobs to each. The work would ultimately be split among the grid's components and the computation would take significantly less time. This was previously tested on an arrangement of grids in *Grid Heterogeneity in In-silico Experiments: An Exploration of Drug Screening using DOCK on Cloud Environments* (W. Yim *et al*) utilizing DOCK 6.2 and the included test suite. Although results were obtained in a much more reasonable amount of time, another problematic instance was discovered when it was determined that not all of the scores matched up with DOCK's developers' scores (see table 1). This is thought to be due to the fact that the tested grids pool the resources of fundamentally different processors with varying operating systems, bits, gcc compiler versions, etc. Yim's publication also explores the possibility of carrying out DOCK experiments in virtualized environments, and finds it to be a useful alternative to grid computing. Not only were the shorter computation times conserved, homogeneity in the networked machines was easily emulated and this yielded the desired consistency in scores.

VMs are created utilizing virtual managing software from a physical node with sufficient storage capacity (what determines a machine to be "capable" depends on the purpose and number of the virtual machines). Several can be hosted from a single server and they can work independently or be networked in a similar arrangement as component machines in a grid just as a physical

machine would. Furthermore, VMs are easily cloned. Cloning a single machine multiple times and networking them into a cluster allows one to simulate a completely homogeneous grid setup of physical nodes. When a small cluster was configured and tested by Yim, scores were better matched with developers' and were obtained within a reasonable time frame.

Objective

This project's purpose consisted of two main objectives:

- Determine a VM configuration that will optimize the molecular docking workflow in pursuit of DuSP specificity inhibitors. Yim found that different virtual machine configurations would have different outputs. To find which combination worked best, several VMs were set up with different OS versions (all of them CentOS), OS bits, and gcc compiler versions. They were all tested using DOCK 6.2's test suite (consisting of a list of ligands, a sample receptor, and the necessary input files) and their performance was assessed by comparing results to developer's scores. The best option was determined by the likeness between the two.
- Compress VM images and upload onto the Pacific Rim Applications and Grid Middleware Assembly (PRAGMA) cloud. The images will be made available to remote hosts such that they can deploy them utilizing the pragma_boot script. Lastly, all instances will be networked utilizing Virtual Private Network's (VPN) N2N application to overcome local firewall settings at each site and maintain VM communication. Once the cluster is set in place, the ligands and receptors necessary for the screening for drug compounds with DuSP-inhibitory properties will be migrated and docked.

Materials and Methods

A series of VMs were created with different machine bit, OS bit, CentOS version, and gcc compiler versions. All were KVM-based and were created with the Virtual Machine Manager application included in the CentOS 6.4 package installed on the physical host. Each VM was cloned once and MPICH 3.4.1 was installed on each instance to enable communication between clones as they would in a cluster. Each cluster was comprised of a "master node" where the user can submit docking jobs from and a "slave node" which would receive the master's command and carry out the computation. After verifying that MPICH and DOCK worked on each cluster, they were tested utilizing the DOCK 6.2 test files. DOCK has two scoring methods: Grid and AMBER based. AMBER based consists of the more realistic assumption that both ligand and receptor could be flexible during the conformations. Grid based assumes a rigid receptor. As a result, grid-based docking takes less time, although it may be slightly less accurate. For our testing purposes, we were mostly interested in comparing grid scores.

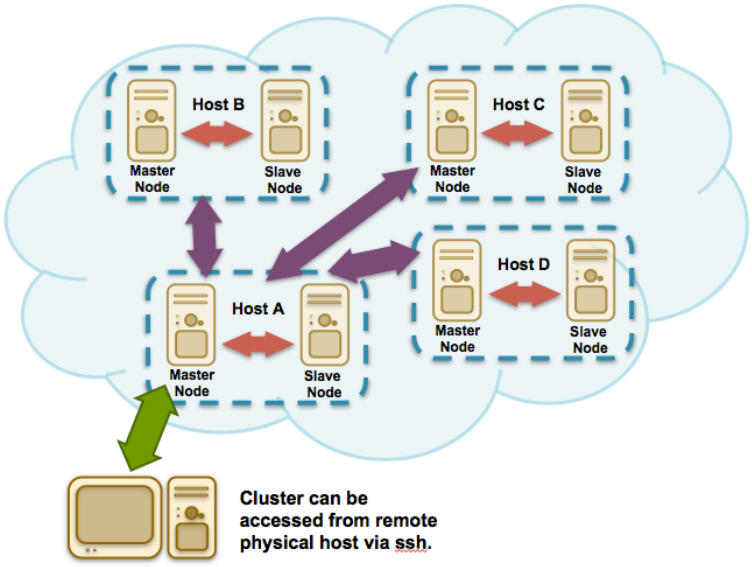

Figure 2 – Schematic of Deployed Cluster to Several Remote Hosts in the PRAGMA Cloud.

The VM pair that matched the developers' scores (also included in the test suite) best was selected to follow through with the project. Once a suitable combination was found, the VM was rebuilt to comply with PRAGMA'S pragma_boot deployment script requirements and with the appropriate OS and compiler versions. The final VM was cloned (such that now there is a master and two slaves), networked, and packaged before being uploaded onto the Nara Institute for Science and

Technology (NAIST) server for testing. The test suite was used again to verify that the VMs worked the same as they did from the physical host.

Master and slave nodes were later packaged and deployed onto the PRAGMA cloud at different sites. Remote SSH login was enabled in the distributed master nodes for secure access and DOCK job submission.

Results

The different VM configurations tested with the DOCK test suite are shown below along with their respective results. Test ligands are listed by their ZINC database ID; all tests were ran against the same receptor and utilizing the same input grid files.

	Developer's	Barco	Master	Sailboat	CentOS1
Machine bit	-	32	64	64	64
OS bit	-	32	32	64	64
CentOS	-	5.9	5.2	5.9	6.4
gcc	-	4.1.2	4.1.2	4.1.2	4.7.7
ZINC00158751	-1.058107	138.11386	138.113861	138.114029	138.114029
ZINC00157960	847.37207	21535.20898	21535.20898	21535.30859	21535.30859
ZINC00158442	52.56588	52.56588	52.56588	52.565788	52.565788
ZINC00013564	10.801939	10.801939	10.801939	-9.400218	-9.400218
ZINC01555236	503.249725	503.249725	503.249725	1800544512	1800544512
ZINC00150863	21.139143	21.139143	21.139143	-12.467216	-12.467216
ZINC00152265	30.238361	30.238361	30.238361	30.238028	30.238028
ZINC00157111	-12.916615	-12.916615	-12.916615	-12.916644	-12.916644
ZINC00157152	-10.137384	-10.137384	-10.137384	-10.137392	-10.137392
ZINC00157402	168513.625	168513.625	168513.625	168506.4063	168506.4063
ZINC00157467	-8.706671	-8.706671	-8.706671	-8.706783	-8.706783

Table 1 – Grid-based scoring of each configuration's results from the DOCK test suite.

As seen in the table above, differences in scoring were appreciable between machines of different OS bits. While there is no sure answer as to why this occurs, it is an issue that DOCK developers are aware of, as they make note of this in the software manual. Additionally, it was concluded that gcc compiler version did not affect results. This was surprising as this was noted in Yim's paper to

be a cause for discrepancy, but this may only be a problem with gcc compilers older than those tested.

Conclusion

Table 1 shows that the best VM configuration would be one with a 32bit operating system and gcc compiler version 4.1.2 or older. This narrowed options to the two configurations named Barco and Master. Out of these two, Barco was selected for two reasons. One was its more recent CentOS version, the other due to the note included in the DOCK manual about the consoles that developers' scores were computed on (a 32bit machine for DOCK 6.2). This VM was then built such that its structure would comply with the requirements of the recently-released PRAGMA deployment script pragma_boot, cloned, and networked as a cluster consisting of a master node and two slave nodes. These instances were deployed and have been distributed to two servers in the PRAGMA cloud: the AIST and UC San Diego hosts. A successful DOCK test was assessed once the machines were uploaded, which yielded results that align with this project's objectives. It is hoped that this cluster can be distributed to additional KVM cluster-supporting sites in the figure, before migrating the entire DuSP inhibitor workflow onto this virtual cluster.

References

1. W. Yim, S. Chien, Y. Kusumoto, S. Date, J. Haga, *Grid Heterogeneity in In-silico Experiments: An Exploration of Drug Screening Using DOCK on Cloud Environments*. Studies in Health Technology and Informatics, 2010: p181-190*

2. Irwin, Sterling, Mysinger, Bolstad and Coleman,*J. Chem. Inf. Model. 2012* DOI: 10.1021/ci3001277

3. P. Lang, D. Moustakas, et al. *DOCK 6.6 User Manual.* University of California, 2013

**Note: Zinc ID's of Dock Test Suite in this document were cut off in the actual publication. Those shown in Table 1 are the same proteins, and the ID's are accurate.*