# Large Scale Execution of a Bioinformatic Application on a Volunteer Grid

Viktors Bertis[6]        Raphaël Bolze [1,3,4]        Frédéric Desprez [2,3,4]

Kevin Reed [5]

[1] CNRS      [2] INRIA      [3] Université de Lyon

[4] LIP laboratory, UMR 5668, ENS Lyon – CNRS – INRIA – UCBL, Lyon, France

[5] IBM Centers for Solution Innovation, Chicago, United States

[6] IBM Systems & Technology Group, Austin, United States

{raphael.bolze,frederic.desprez}@ens-lyon.fr

{bertis,knreed}@us.ibm.com

**Abstract:** Large volunteer desktop platforms are now available for several applications. This paper presents the work that we did to prepare the first phase of the Help Cure Muscular Dystrophy project to run on *World Community Grid*. The project was launched on December 19, 2006, and took 26 weeks to complete. We present performance evaluation of the overall execution and compare a volunteer grid with a dedicated one.

**Keywords :** *desktop computing, docking application, World Community Grid, grid performance evaluation, grids comparison.*

## 1   Introduction

Large scale computing platforms based on internet-connected PCs volunteered by their owners are now widely used for many applications over the world. These applications range from the search to extraterrestrial intelligence [4] to bioinformatic applications [10]. Those loosely coupled applications behave nicely on such large platforms and the aggregated computing power can be compared to huge supercomputers only available in large computing centers, but at a lower cost. However, this performance comes at a cost, the volatility of the nodes that leads to use of fault tolerant algorithms, mostly based on computation replication.

*World Community Grid*'s mission is to create a large public computing grid to help projects with high social impacts [9]. *World Community Grid* has built a grid platform using a commercial solution from UNIVA-UD [12] and also uses the BOINC open source solution. The BOINC system [3] is one of the most well known systems designed to build large scale volunteer computing system. Its simple and robust design allows the development of large scale systems for several important applications all over the world.

In this paper, we present our experience in porting a docking application, which predicts protein-protein interactions, to *World Community Grid*. We are especially interested in performance estimation and the way we can predict the execution time for larger problems on such a large scale grid.

Several papers have presented such approaches for similar applications. In [10], the authors present their experience of the use of P@H/BOINC system for testing structure prediction algorithms based on conformational sampling. In [13], clustered processors are put within the United Device MetaProcessor to allow task and data parallelism to be used in the parallel CHARMM application. Their target application is protein folding.

The rest of this paper is organized as follow. The next section presents our target application, i.e. the "Help Cure Muscular Dystrophy project". Section 3 gives details about *World Community Grid* which we used to solve our problem. Section 4 gives details about the way the application is sliced in workunits of a given size that allow the best performance on *World Community Grid*. Then, we describe the execution of the whole application over the grid in section 5. Finally, before the conclusion, we present the performance comparison of the Desktop Grid Platform with a dedicated grid for this specific application.

## 2 Help Cure Muscular Dystrophy project (HCMD)

This project has been carried out in the framework of the Décrypthon program, set up by CNRS (Centre National de la recherche Scientifique), AFM (French Muscular Distrophy Association), and IBM. This project proposes to use the power offered by grid computing to the detection of protein-protein interactions. It directly addresses these questions by setting the goal of screening a database containing thousands of proteins for functional sites involved in binding to other proteins targets. Information obtained on the structure of macromolecular complexes is important not only to identify functionally important partners, but also to determine how such interactions will be perturbed by natural or engineered site mutations in either of the interacting partners, or as the result of exogenous molecules, and notably, pharmacophores. A database of such information would be of significant medical interest since, while it now becomes feasible to design a small molecule to inhibit or enhance the binding of a given macromolecule to a given partner, it is much more difficult to know how the same small molecule could directly or indirectly influence other existing interactions.

The development of a docking algorithm to predict protein-protein interactions exploits knowledge of the location of binding sites. The basic algorithm uses a reduced model of the interacting macromolecules and a simplified interaction energy. Optimal interaction geometries will be searched for using multiple energy minimizations with a regular array of starting positions and orientations. Later on, knowledge of binding sites will greatly reduce the costs of the search. It provides the basis for real-case predictions of functionally significant partners. Further information on this project can be found on the web site at *World Community Grid*.

### 2.1 MAXDo program

The MAXDo program (Molecular Association via Cross Docking simulations) has been developed by Sacquin-Mora et al. [8] for the systematic study of molecular interaction within a protein database. The aim of the docking process is to find the best way to associate two proteins in order to form a protein-protein complex (and see whether these two proteins are likely to interact, should they ever meet in a biological system). The quality of the protein-protein interaction can be evaluated through an interaction energy (expressed in $kcal.mol^{-1}$), which is the sum of two contributions; a Lennard-Jones term ($E_{lj}$), and an electrostatic term ($E_{elec}$), which depends on the electric charges that are located all over the protein. The more negative the sum of these two contributions is, the stronger the protein-protein interaction. The MAXDo program uses a reduced protein model developed by M. Zacharias [14]. The proteins are rigid and the minimization of the interaction energy is computed according 6 variables: the space coordinates $x, y, z$ of the mass center of the ligand and the orientation of the ligand $\alpha, \beta, \gamma$. Those degrees of liberty are concatenated into 2 parameters: $i_{sep}$ and $i_{rot}$ which respectively represents the *starting position* $(x, y, z)$ of the mobile protein $p_2$ (also called the ligand) with respect to the fixed protein $p_1$ (called the receptor), and the *starting orientation* $(\alpha, \beta, \gamma)$ of protein $p_2$ relatively to the fixed protein $p_1$. Then minimizing the interaction energy between two proteins for a set of initial positions and orientations of the ligand gives a map of the interaction energy for the pair proteins.

More formally, the MAXDo program is defined for one pair of protein $(p_1, p_2)$ as the computation of the interaction energy between them:

$$E_{tot}(i_{sep}, i_{rot}, p_1, p_2)$$

where $i_{sep}$ is the starting position and $i_{rot}$ is the starting orientation of the ligand with respect to the receptor. So the map of the interaction is obtained by computing the set of docking energy:

$$\forall i_{\text{sep}} \in [1..N_{\text{sep}}(p_1)], \forall i_{rot} \in [1..N_{\text{rot}}], E_{tot}(i_{\text{sep}}, i_{\text{rot}}, p_1, p_2)$$

Notice the number of starting positions between the two proteins depends on the receptor $p_1$ and this number is directly linked with the size and shape of the protein. Finally, the total number of MAXDo program's instances which have to be launched for a set of proteins $P$ is defined by this quantity:

$$\forall (p_1, p_2) \in P, \forall i_{\text{sep}} \in [1..N_{\text{sep}}(p_1)],$$
$$\forall i_{rot} \in [1..N_{\text{rot}}], E_{tot}(i_{\text{sep}}, i_{\text{rot}}, p_1, p_2)$$

The first phase of the HCMD project targets 168 proteins, the number of rotations have been fixed by the scientists $N_{\text{rot}} = 21$[1], and the starting positions are evaluated by an other program for each protein. The MAXDo program is not symmetric, i.e. $if p_1 \neq p_2, E_{tot}(i_{\text{sep}}, i_{\text{rot}}, p_1, p_2) \neq E_{tot}(i_{\text{sep}}, i_{\text{rot}}, p_2, p_1)$. These 168 proteins have been selected because they are all known to take part in at least one identified protein-protein complex and they cover a wide range of protein structures and functions without redundancy [7].

This project follows a first study on 6 proteins which was performed on the dedicated grid of the Décrypthon project.

---

[1]In fact the number of *starting orientation* is 210 : 21 couples $(\alpha, \beta)$ for 10 values of $\gamma$.

This study argues that preliminary work showed that the docking program required a lot of *CPU* time and produced promising scientific results [8] and will take advantages of desktop grid computing.

## 3  *World Community Grid*: a volunteer grid

Launched November 16, 2004, their sponsor, IBM, provides the technical infrastructure, expertise, and hosting to support *World Community Grid*. The scientific projects are selected by an advisory board composed of prominent philanthropists, scientists, and officials from leading public and private organizations and reviews proposals to identify those with the best potential to benefit from technology of *World Community Grid* and make important progress on humanitarian goals [9]. This type of grid encourages public awareness of current scientific research, it catalyzes global communities centered around scientific interest, and it gives the public a measure of control over the directions of humanitarian scientific progress.

### 3.1  Desktop grid description

*World Community Grid* is a volunteer grid, also known as a Desktop grid, or distributed grid. Basically this grid is composed of several servers that host a database of computing work (data + program) named *"workunit"*. The volunteers have to register to the web server at *World Community Grid* and download an agent which will be responsible to contact the workunits database. This agent is also in charge of monitoring and checking workunit computations. The agent connects to the server to get new workunit, then it launches the program with the specific parameters corresponding to the piece of work they have been selected for. After the computing work is finished, the computing device sends back the result to *World Community Grid* and asks for an another workunit. The user can configure the agent to use only the idle time of the device, or launch the workunit only when the screensaver is active or continuously work for the *World Community Grid* projects.

*World Community Grid* is using two different types of middleware in order to provide the infrastructure of the Desktop grid: the BOINC system [2] and the Univaud's middleware Grid MP [12]. Actually, there are more than 344,000 subscribed members and more than 836,000 declared devices. You can subscribe several devices with the same member profile. The web site at *World Community Grid* provides some basic information about the status and the global statistics of their grid [11]. According to this information we can draw the graph of the number of *virtual full-time processors* which participate in *World Community*
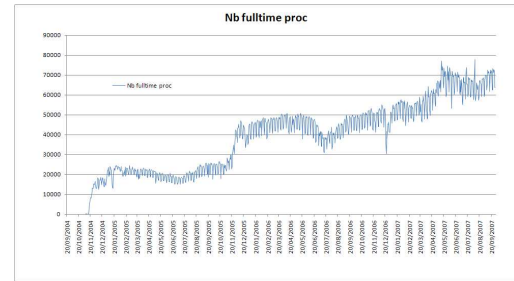


**Figure 1.** *virtual full-time processors* **of** *World Community Grid*

*Grid*. With his new paradigm of *virtual full-time processors*, we answer the question: "How many processor do we need to generate 10 years of *CPU* time for 1 day ?". If for 1 day, 10 years of *CPU* time are consumed, it is equivalent to at least 3 650 processors that compute full time for 1 day. This notion of *virtual full-time processors* does not say anything about the processor. We do not have any information about the power of the grid and we only know the minimum number of processors needed to generate the *CPU* time. In our sense, the notion is easier to understand than the total *CPU* time and it gives a better idea of how large is *World Community Grid*. We will use this paradigm in order to compare *World Community Grid* with a dedicated grid.

Figure 1 shows the evolution of the number of *virtual full-time processors* which participate in the project hosted by *World Community Grid* since the beginning of the grid. The web site at *World Community Grid* provides the total *CPU* time generated by days. We convert this value into *virtual full-time processors*. We can notice the number of *virtual full-time processors* globally increases. The curve is not regular because during the week-end there are less processors than during the week. There are some periods where the number of processors went down; Christmas holiday of 2005 and 2006 and summer time of 2006.

### 3.2  Needs and requirement

Scientific projects must meet three basic technological requirements to ensure benefits from *World Community Grid* computing power (this requirements are given in the documents called "Request for proposals" [1]). Projects should have a need for millions of *CPU* hours of computation to proceed. The computer software algorithms required to accomplish the computations should be such that they can be subdivided into many smaller independent computations. And finally, if very large amounts of data are required, there should also be a way to partition the data into sufficiently

small units corresponding to the computations.

These constraints are directly related to the nature of *World Community Grid*. The computers which compose the membership of *World Community Grid* are usually simple desktop machines. Desktop computers have more and more computing and storage capacities, fast internet connections, but the work distributed to the volunteer computer has to need reasonable memory space and data transfer time. Empirically, the team at *World Community Grid* has determined a workunit should last around 10 hours. This value can be considered as a human factor, it represents the time a volunteer would wait to accomplish a workunit, it is also a good value to monitor the progression of the work. This value is also constrained by the capacity of the servers at *World Community Grid* to distribute the work to volunteers device. It determines the rate of transactions with *World Community Grid* servers. An interesting study on performances issue of a BOINC task server have been done by the BOINC team [5].

## 4 Workunits preparation

As mentioned in the requirements for *World Community Grid*, the work should be partitioned into small pieces of work that ideally takes 10 hours to complete. In order to be able to achieve this goal we have to estimate the computing time needed by the MAXDo program.The first evaluation determined the parameters $N_{\text{sep}}(p)$ for the 168 proteins. This parameter is the number of starting positions of a ligand around a given receptor.Then important properties of the MAXDo program computing time $ct(i_{\text{sep}}, i_{\text{rot}}, p_1, p_2)$ have been established: (1)The MAXDo program has a reproducible computing time.(2) For one pair of proteins $p_1, p_2$, if the parameter $i_{\text{sep}}$ is fixed, then the MAXDo program is linear in the parameter $i_{\text{rot}}$. (3) For one pair of proteins $p_1, p_2$, if the parameter $i_{\text{rot}}$ is fixed, then the MAXDo program is linear in the parameter $i_{\text{sep}}$.

We launched the MAXDo program on four clusters with similar nodes (i.e. dual Opteron 246 @ 2 Ghz) on the Grid'5000 [6] platform. 640 processors were used for this experiment during one day. This experimental run gives us the complete matrix $M_{ct}$ of computing time where the entry $ct_{i,j}$ represents the computing time for the pair of proteins $(p_i, p_j)$. This $168^2$ run consumed more than 73 days of *CPU* time, The matrix $M_{ct}$ and the $N_{\text{sep}}$ table offer the possibility to evaluate the total *CPU* time needed on the reference processor. It needs more than 14 centuries and 88 years of *CPU* time on a single Opteron 2Ghz processor to be precise 1,488:237:19:45:54 (y:d:h:m:s). This quantity is represented by formula:

$$\sum_{p_1, p_2 \in P} N_{\text{sep}}(p_1) * 21 * ct_{\text{iter}}(p_1, p_2) \qquad (1)$$

Where $ct_{\text{iter}}(p_1, p_2)$ represents the entry value in $M_{ct}$ for the pair of proteins $(p_1, p_2)$; i.e time needed for the MAXDo program to run $E_{tot}(1, 1, p_2, p_1)$ and $N_{\text{sep}}(p_1)$ represents the value in the $N_{\text{sep}}$ table, which give the number of starting positions around the protein $p_1$.

At this time, some observations can be pointed out. The total needed *CPU* time on the reference *CPU* is huge 1,488:237:19:45:54 (y:d:h:m:s). The MAXDo program is embarrassingly parallel : each step can be computed independently from every other, and there is a huge number of steps (49,481,544 workunits can be generated). The data needed for the MAXDo program is small: the 2 proteins files + program + parameters (no more than 2 MB).

These observations show that the MAXDo program is a perfect candidate for a distributed grid such as *World Community Grid*.

### 4.1 Workunit packaging

The objective is now to slice the whole work represented by the formula (1) into small pieces of work which lasts approximately $h$ hours ($h \simeq 10$). The constraints are the following. For a given pair of proteins $(p_1, p_2)$, only the number of the $N_{\text{sep}}$ to compute can change and must remain in interval $[1..N_{\text{sep}}(p_1)]$, the number of orientations $N_{\text{rot}}$ is fixed to 21. The workunit is defined for one pair of proteins, i.e we cannot build a workunit with three proteins $p_1, p_2, p_3$ in order to compute on a piece of the work of $(p_1, p_2)$, then $(p_2, p_3)$ and so on.

The two previous constraints are technical because it will demand unnecessary additional work to merge result files. So the problem is to find the parameter $n_{\text{sep}}$ which is the number of separation points to compute in one workunit for one pair of protein,
$\forall (p_1, p_2) \in P$, find $n_{\text{sep}} \in [1..N_{\text{sep}}(p_1)]$,

if $\quad \left\lfloor \frac{h}{21.ct_{\text{iter}}(p_1,p_2)} \right\rfloor \leq 1, \qquad n_{\text{sep}} = 1$

if $\quad \left\lfloor \frac{h}{21.ct_{\text{iter}}(p_1,p_2)} \right\rfloor \geq N_{\text{sep}}(p_1), \quad n_{\text{sep}} = N_{\text{sep}}(p_1)$

$\qquad \text{else} \qquad n_{\text{sep}} = \left\lfloor \frac{h}{21.ct_{\text{iter}}(p_1,p_2)} \right\rfloor$

There are several methods to build workunits. We can have sub goal such as to decrease the number of small workunits or minimize the number of workunits. It also depends on the softness of the $h$ parameter.
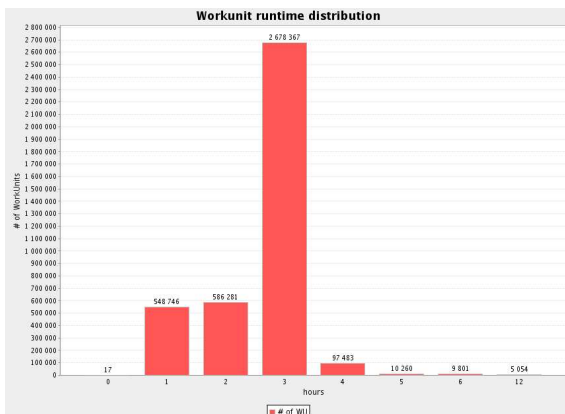
4

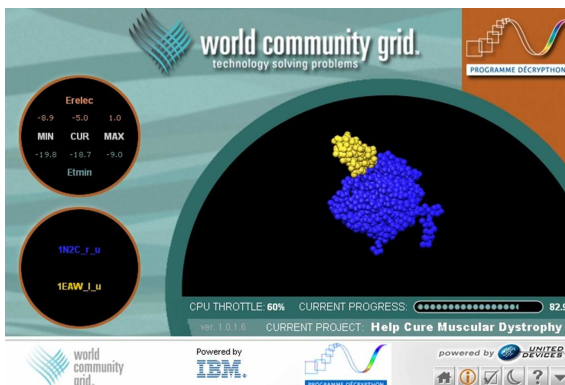**Figure 2. Real workunit distribution.**



**Figure 3. HCMD screensaver.**

Figure 2 shows the distribution of real workunits which were sent to the volunteers Most of workunits were tuned to take between 3 and 4 hours of *CPU* time.

## 4.2 Porting to *World Community Grid*

The MAXDo program has been packaged into a program with a screensaver and modified in order to monitor the progression of the program. This job has been done by the technical team at *World Community Grid*. Figure 3 shows the screensaver of the HCMD project. Several pieces of information are shown in the MAXDo agent: the name and the graphic of the two proteins which are currently being docked, the value of the docking energies, the current progress of the docking program, and the links to the website of the different partners of the HCMD project.

Furthermore, the technical team adds a checkpoint feature to the MAXDo program. The MAXDo program can be stopped at any time and restarted from the last checkpoint.

This feature is essential; as the volunteers can stop or kill the process at any time, checkpoints are essential to preserve computation that have been carried out. Anyway the checkpoint occurs only between *starting positions*. If the program is stopped during the computation of one starting position, the MAXDo program has to be relaunched from this position.

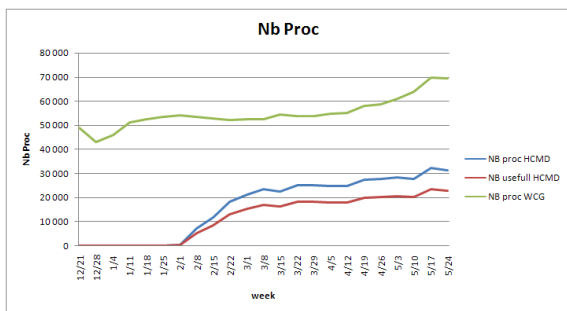## 5 HCMD project launched on *World Community Grid*

### 5.1 Computing phases

This previous section explains how the MAXDo program and the target set of proteins were tuned to fit to the constraints of *World Community Grid*. The computation phase was launched on December, 19, 2006. The *World Community Grid* team decided to launch the workunit of one protein after an other. They also decided to first launch the proteins that required less computing time. This choice was motivated by the fact that it can be easier to detect the failures on the beginning of the project when results are returned quickly from the volunteers. Furthermore *World Community Grid* is dynamic, there are always new members that join the grid with brand new machines. So these new faster devices can work on more time consuming workunits.
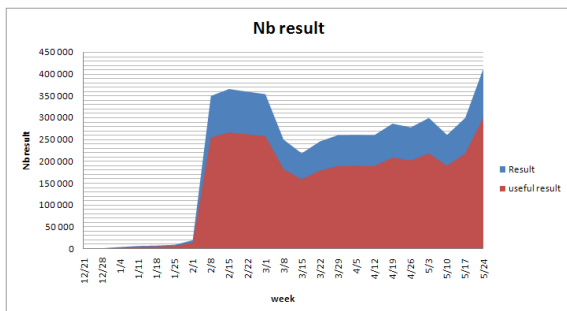
Figure 4(a) shows the number of processors that participated to the HCMD project. This graphic is generated with the data provided by the *World Community Grid* team. They give us the *CPU* time in years consumed each week. With this value, we estimate the number of *virtual full-time processors* that participate to HCMD. Then the following remarks can be given on Figure 4(a):

During the project the number of processors that participate in *World Community Grid* always increased. The average number of processors available is 54,947, the average number of processors dedicated to the HCMD project is 16,450.

Three different periods can be distinguished for the HCMD project : (a) During the two first months of the project. there are just a few processors that participated in the HCMD project. The priority of this project was very low compared to the others. This phase can be assimilated to the "*control period*". (b) In February, the number of processors which participated to the HCMD project increased. At the end of February, 45% of *World Community Grid*'s devices participated to the HCMD project. This phase can be named "*project prioritization*". (c) From march to the end of the HCMD project: 4 months, the part of the processors which participate to the HCMD project stayed constant.

(a) # of *virtual full-time processors* during the HCMD project.



(b) # of results during the HCMD project.

**Figure 4. HCMD project on** *World Community Grid***.**

As the number of processors in *World Community Grid* increased, consequently the number of processors dedicated to the project increased. The phase can be named "*full power working phase*". During this phase 26248 processors were used on average.

Figure 4(b) shows the number of results received during the project. The number of results is not necessarily linked with the number of processors (Figure 4(a)), i.e. the *CPU* time consumed, because each result does not represent the same amount of work. There are two areas in Figure 4(b), only 73 % are useful results. *World Community Grid* system sends more than one copy of each workunit to the volunteers. This is call "redundant computing". This mechanism allows *World Community Grid* to identify and reject erroneous results. Redundant computing was used in two cases: the result returned is not correct or the workunit sent to a volunteer reached the timeout. As the grid is composed of volunteers, we can expect that there are some people who do not connect to the *World Community Grid* servers for a long period, then when the agent reconnects and sends back the result to the servers, this results is taken into account even if the result has already been computed by some other device. The redundancy factor for the project is 1.37, it
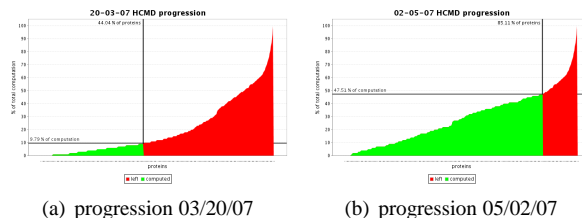


(a) progression 03/20/07     (b) progression 05/02/07

**Figure 5. HCMD project progression**

is obtained by comparing the number of computing results disclosed by *World Community Grid* (5,418,010) and the number of effective results received from *World Community Grid* (3,936,010). This factor was not constant during all phases of the project. It was higher at the beginning, because the results were compared to each other to be validated, but later we provided a method to validate the results by checking the values returned in the result file (there are some specific boundary conditions on each value).

## 5.2 Result processing and verification

During the project, the *World Community Grid* team sent results that were calculated by the volunteers to a storage server in France. Then we were in charge of validating those results. The output of the MAXDo program is a simple text file that contains on each line the coordinate of the ligand and its orientation, and then the interaction energies values. The *World Community Grid* team sent us the results when one protein has been docked with the 168 others. Each time we received the results, we validated those results with 3 different checks: check if there are the correct number of files, check if there are the correct number of lines in the files, and check if the values in the file are within a valid range. Then when the files were checked, we merged result files in order to have one result file for one pair of proteins. All these result files represents 123 Gb of text files ( 45 Gb compressed) and there are $168^2$ files.

In addition to these controls, we provide the graphics shown in Figure 5 which represents the progression of the project. The proteins are on the X axis, and the Y axis represents the cumulative percentage of computation. The light part (on the left) is the percentage that has been computed, the dark part (on the right) is the not yet computed part. This graphic effectively shows that the time needed for each protein is different. For example, on the 05-02-07 (i.e 5 month after the beginning of the project) 85% of the proteins were docked, but this represents only 47% of the 1488:237:19:45:54 (y:d:h:m:s) total computation.

6

## 6 Comparative performance with a dedicated grid

The *World Community Grid* web site reports the total *CPU* time consumed by the HCMD project is 8,082:275:17:15:44 (y:d:h:m:s). This quantity is 5.43 times higher than the estimated value with the reference processor Opteron 2 GHz on the Grid'5000 platform. The total *CPU* time includes the redundant workunits (the redundancy factor is 1.37). If we take into account this fact, the speed down is 3.96.

Furthermore the distribution of real workunits which were sent to the volunteers reported in Figure 2, confirmed the speed down factor. This distribution shows that most workunits were tuned to take between 3 and 4 hours of *CPU* time (average is 3 hours 18 min 47s), according to the evaluation matrix obtained by the experiment on the reference processor opteron@2Ghz. For the whole project the average *CPU* time obtained on *World Community Grid* device is around 13 hours (obtained with the total number of results and the total *CPU* time). This confirms the speed down value 3.96 (13 hours / 3.96 = 3h15): the average workunit computing time obtained with the packaging is approximately 4 times less than the real average execution time.

Several arguments can be given to explain the speed down factor. The first comment is that there are some limitations to the way that the UD agent accounts for run time. In particular, the UD agent measures wall clock time rather then actual process execution time. This comes into play in a couple of ways. First of all, *World Community Grid* has set the work for the UD agent to run at most at 60% of *CPU* time. This can only be changed by downloading a separate utility from the *World Community Grid* website to change this setting. This means that a computer using the UD agent with the 60% *CPU* throttle that runs a workunit for 8 hours of "wall clock" time will at most only actually process work for 4.8 hours. Secondly, since the research application runs at the lowest priority on the computer, any other use of the computer's processor will further reduce the actual amount of time that the research runs. This can further reduce the actual *CPU* time. In essence it would not be unexpected if the research application actually ran for less then 50% of the elapsed wall clock time. This means that run time reported overstates that actual amount of time the research was running. These items can explain about half of the 3.96 value. Secondly, the devices on *World Community Grid* are not dedicated. It means that the volunteers share their computer, the members can stop, kill or nice the *World Community Grid* agent if they are disturbed by the program. In that case, the program is relaunched by the *World Community Grid* agent at the last checkpoint, and these interruptions

| Grid | whole period | full power |
|---|---|---|
| *World Community Grid* | 16,450 | 26,248 |
| Dedicated Grid | 3,029 | 4,833 |

**Table 1. Table of equivalence between** *virtual full-time processors* **of** *World Community Grid* **and processors of dedicated grid**

consumed a large part of the additional computing time. In addition the devices on *World Community Grid* are slower (on average) than an Opteron 2 GHz, and screensaver itself can add to *CPU* usage in varying degrees, depending on platform and speed of the machine. Nevertheless, even if the performance of volunteers device is quite low comparing to the power of one processor of a dedicated grid, this weakness is balanced by the huge number of *virtual full-time processors* of this kind of grid.

Table 1 represents the equivalence[2] between the average number of *virtual full-time processors* which were consumed during the HCMD project and the number of processors which would be necessary on a dedicated grid such as Grid'5000. Two distinct periods are shown, the *whole period*, i.e. from the beginning to the end of the project, i.e. six months, or the *full power working phase* which is only the four months where the HCMD project had a high priority on *World Community Grid*.

To be specific, during the prior week that this paper was written, *World Community Grid* received 1,435 years of run time or an average of 74,825 days of run time per day. This equates to 74,825 *virtual full-time processors*. Using the factor 3.96 determined above, this suggests that *World Community Grid* currently provides at least the equivalent of 18,895 Opteron 2 GHz processors on a Grid'5000. This is a low estimate of *World Community Grid*'s computing power as BOINC measures run time more accurately then UD. We will explore this idea further in a future paper.

## 7 Conclusion and future work

Large scale executions of applications on volunteer grids are no longer research projects. Several applications have been successfully ported on such large scale platforms. Among them, BOINC is one of the most used system due to its performance and ease of use.

Our target docking application fits nicely on such a platform. The HCMD project was launched on December 19, 2006 and finished on June 11, 2007. It took 26 weeks to

---

[2]This comparison has to be taken carefully, since it supposed that the dedicated grid is optimally used.

complete. During this time, 168 proteins were docked 2 by 2 that generates 123 Gb of data and consumed more than 80 centuries of *CPU* time.

In our paper, we described all the steps which were needed to be able to launch this large scale execution for our bioinformatic application. We showed the benefits of using a volunteers grid, but also described the limitations of this kind of grid compared to a dedicated one. The run of this project require 5.43 times more *CPU* time than expected. We have introduced the notion of *virtual full-time processors* which characterizes a volunteer grid against a dedicated one. We showed the *virtual full-time processors* of *World Community Grid* is around 4 times slower than a opteron @2Ghz processor.

Additionally in phase I of the HCMD project, the MAXDo program was only run on the UD agent, but in phase II the program will only be run on the BOINC agent. There exists differences between the way the two middleware systems account for run-time which may introduce differences in what represents a *virtual full-time processors*. Another way to approach the number of *virtual full-time processors* is to base the estimate on the number of points awarded instead of run-time. Points represent the amount of work done by computer to compute a result and are based on the run time for that result multiplied by a weight factor determined by running a benchmark on the agent. This approach should reduce the differences between each platform therefore be more middleware independent. This approach should also allow us to observe the trend toward more powerful processors in desktop computers.

Eventually, this project is a perfect example of how the specificities of two different kinds of grid (dedicated and volunteer) can be used in order to solve a bioinformatic project which required a huge amount of computation.

## Acknowledgments

## References

[1] W. C. G. advisory board. World Community Grid request for proposals. `http://www.worldcommunitygrid.org/bg/rfp.pdf`.

[2] D. Anderson. BOINC: A System for Public Resource Computing and Storage. In *Proceedings on the Fifth IEEE/ACM International Workshop on Grid Computing (CGRID04)*, 2004.

[3] D. Anderson, C. Christensen, and B. Allen. Designing a Runtime System for Volunteer Computing. In *Proceedings of the Supercomputing Conference*, Tampa, Florida, USA, 2006.

[4] D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. SETI@home: An Experiment in Public-Resource Computing. *Communication of the ACM*, 45(11), 2002.

[5] D. Anderson, E. Korpela, and R. Walton. High-performance task distribution for volunteer computing. In *E-SCIENCE '05: Proceedings of the First International Conference on e-Science and Grid Computing*, pages 196–203, Washington, DC, USA, 2005. IEEE Computer Society.

[6] R. Bolze, F. Cappello, E. Caron, M. Daydé, F. Desprez, E. Jeannot, Y. Jégou, S. Lantéri, J. Leduc, N. Melab, G. Mornet, R. Namyst, P. Primet, B. Quetier, O. Richard, E.-G. Talbi, and I. Touche. Grid'5000: a large scale and highly reconfigurable experimental grid testbed. *International Journal of High Performance Computing Applications*, 20(4):481–494, Nov. 2006.

[7] J. Mintseris, K. Wiehe, B. Pierce, R. Anderson, R. Chen, J. Janin, and Z. Weng. Protein-protein docking benchmark 2.0: An update. *Proteins: Structure, Function, and Bioinformatics*, 60(2):214–216, 2005.

[8] S. Sacquin-Mora, A. Carbone, and R. Lavery. Identification of protein interaction partners and protein-protein interaction sites via cross-docking simulations. *to be published*.

[9] W. C. G. staff. World Community Grid web site. `http://www.worldcommunitygrid.org`.

[10] M. Taufer, C. An, A. Kerstens, and C. B. III. Predictor@Home: A Protein Structure Prediction Supercomputer Based on Public-Resource Computing. *IEEE Transactions on Parallel and Distributed Systems*, 2006.

[11] W. C. G. team. World Community Grid global statistics page. `http://www.worldcommunitygrid.org/stat/viewGlobal.do`.

[12] U. UD. Univa UD, PCs grid solution: Grid MP. `http://www.univaud.com/products/grid-mp/`.

[13] B. Uk, M. Taufer, T. Stricker, G. Settanni, A. Cavalli, and A. Caflisch. Combining Task- and Data Parallelism to Speed up Protein Folding on a Desktop Grid Platform - Is efficient protein folding possible with CHARMM on the United Devices MetaProcessor. In *Proc. of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid '03)*, May 2003.

[14] M. Zacharias. Protein-protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Sci*, 12(6):1271–1282, 2003.