Theoretical Study of Replication in Desktop Grid Computing: Minimizing the Mean Cost

Ilia Chernov
Institute of Applied Math Research, Kareliam Research Centre of RAS
185910 Pushkinskaya 11
Petrozavodsk, Russia
chernov@krc.karelia.ru

ABSTRACT

We consider a model of computing process with independently produced results. Enterprise desktop grid is kept in mind as a computing tool. All nodes are equal. Possibility of producing wrong results is taken into account. We assume that a priori distribution of possible answers and probabilities of producing one answer while another is correct are known. In case a wrong result is accepted, some penalty is added to the computation cost. m-first voting replication scheme is used to minimize the overall average expenses. A task is replicated until a given number of identical answers are obtained. The problem is to choose the optimal quorums which can depend on answers. We consider the most general model and show how to solve the optimization problem. A few simple and asymptotic cases are studied. The main conclusion is that optimal quorums are quite stable with respect to penalties, so there is no need to know their exact values. Also we consider two groups of computing nodes and show on an example that well-chosen replication scheme on weaker computers can be better than using faster ones.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of systems—*Fault tolerance*; C.2.4 [**Computer Systems Organization**]: Performance of systems—*Distributed systems*

General Terms

Theory

Keywords

desktop grid, replication, reliability, volunteer computing, optimal quorum

1. INTRODUCTION

Desktop grids have become a cheap and rather powerful tool for solving various problems from different branches of science. Enterprise desktop grids (EDG, see, e.g., [1]) gather desktop computers, servers, and other resources from one or a few institutions to solve multiple computing tasks connecting via LAN or Internet. EDG does not suffer from malicious actions and unpredictable switch-offs; However, wrong results can be returned due to a number of reasons. This can be malfunction of hardware, corruption of data, algorithmic errors, wrong results produced by the correct algorithm. An example of the latter case is possible convergence of a descent method to false minima.

Most errors can be revealed by replication: solving the same problem a few times on different computers. We use the quorum approach, also called m-first voting: the result task is replicated until ν identical results are obtained to be accepted as the truth. Of course, the quorum ν can depend on the answer, i.e., some answers can be checked more carefully. Note that this approach differs from the majority voting where the answer received most times from N replicas is believed to be true: in this case redundancy is fixed, while in the m-first voting number of replicas can grow quite significantly; on the other hand, majority voting can fail to produce a result and may lack reliability.

Choosing the quorum values is not an easy problem. If too high, much resources would do useless work, though underestimation is too risky. We minimize the average cost of computation per a task as expected expenses on solving a task a few times with expected pelaties paid in case a wrong answer has been accepted. To the best of our knowledge such approach of minimizing the total mean cost including spent time and possible penalties has not been considered so far.

Let us consider virtual screening [2] as an example. Software is able to evaluate energy of binding between a small molecule called ligand and a larger protein molecule. Calculation can be performed using different parameters, precision and other, so false solutions sometimes appear. Algorithms use random numbers, therefore, a few tries even on similar computers are able to reveal the mistake. In case such wrong answer is accepted, the substance is recommended for biochemical test in a laboratory, which is rather costly; this cost together with reputation losses, pointless use of computing resources, etc, forms rather high, compared to the cost of a single computation, penalty.

A drawback of such approach is lack of precise information about the penalty values. It is by no means easy to estimate losses in case of producing a wrong answer in terms of cost of an individual task, especially in heterogeneous Desktop grids. However, as we show, rough estimation is enough:

ICAIT '16, Oct. 6 – 8, 2016, Aizu-Wakamatsu, Japan.

optimal quorums are rather stable with respect to penalty values, at least in practically important case of low risk.

Beside solving the problem of optimal replication given risk levels for individual calculation and penalty threats, one can consider another problem: to choose penalties that force the desired replication (at least given or higher) with minimal possible average cost.

Although we focus on EDG, the presented approach can be applied also to volunteer computing systems, because the concept of probability of the correct answer can be used, with some restrictions, to counteracting saboteurs also.

The rest of the article is organized as follows. We review the related work, then propose the model, as general as possible, and derive formulae for probabilities and the mean cost that can be used in practice. Also we give hints on how to solve the optimization problem. We consider a few simple cases: one with an absolutely reliable answer and a recognition problem. Here we are able to solve the optimization problem analytically. We define the concept of *critical penalties* that force change of quorums; typically they are quite rare, so there is no need to know the penalty exactly. This fact is important for practical use of our approach. We develop this idea by considering asymptotic analysis of the model in case of low risk. Also we show that quicker (or cheaper) but less reliable computers can over-compete slower (more expensive) at the same level of risk (and higher redundancy). In the conclusion plans for future are described.

2. RELATED WORK

By replication in this article we mean always task replication. Replication has been used in desktop grid computing since its creation: BOINC middleware [3] supports redundant computing to identify wrong results. The survey [4] reviews fault-tolerance techniques, including job replication; though other methods discussed there hardly can help to reveal mistakes: they reduce losses due to unexpected switch-offs, unfinished tasks, etc.

Beside increasing reliability of the system, replication is often used for optimizing productivity of the computing system using a few metrics, when it is important to complete a task as soon as possible, even paying the cost of redundancy. One of the metrics is makespan: time for completing all tasks. For example, [5] considers a typical EDG computing situation with rather low amount of tasks; so near the end of computing process there are free resources while each error or switch-off drastically slows down the whole process. Duplication of tasks improves the makespan.

In [6, 7] replication is studied theoretically from the point of view of improving productivity of multicomputer system. The authors of [8] consider how to reduce loss of computation power due to replication by choosing optimal replication according to nodes' reliability. In [9] replication is used to reduce risk of violating deadlines in case of unreliable computers. Article [10] considers the problem from a different point: sometimes it is necessary to run replicas on absolutely identical computers due to strong dependence of results on precision and other technical details of computing process.

Much attention has been payed to replication as one of the methods ([11] is a review of them) for revealing malicious actions (sabotage) in volunteer computing systems.

The work [12] studies errors is Internet computing grids. It provides some statistically obtained estimates: about one third of hosts return a wrong result at least once; the mean error rate is about 0.002; majority voting is shown to be able to reduce risk to $2 \cdot 10^{-4}$. Also errors from hosts showed no correlation: this allows to extend our approach to Internet computing also.

3. THE MATHEMATICAL MODEL

3.1 Assumptions

Let x_i, $1 \le i \le I$ be the set of possible answers to some problem. The nature of x_i does not matter: this can be real or integer numbers, matrices, vectors, texts, boolean values, etc. However, some knowledge is usually available about distribution of x_i: some are more expected than others; let us denote the *a priori* distribution of x_i by α_i.

Answers are produced by some computing system (we keep in mind an Enterprise desktop grid). What is important for the model is only the fact that different answers are produced independently.

As we have noted above, there is often probability of mistake: producing a wrong answer in desktop grid computing. The reasons can be different, as we have noted above. The probability of error can, in general, depend on what answer is indeed correct. So let us denote the probability of getting the answer x_i while x_j is true by p_{ij}. In the absolutely reliable system p_{ij} is the Kronecker delta-symbol.

In case of accepting a wrong answer we suffer some penalty, either by some kind of direct fine, or loosing reputation, spending funds on needless examining the object in lab, doing much needless work, etc. Let us denote the penalty paid in case of accepting x_i while x_j is correct by F_{ij}.

All computing nodes are identical. Let the cost of producing an answer s_i is C_i. In the end of the article we consider two groups of computing nodes of different reliability and efficiency.

So, the *a priori* distribution α_i of the answers, conditional probabilities p_{ij} of getting an answer x_i while another answer x_j is correct, penalty values F_{ij}, and costs C_i are known. As we have already noted and will prove in the sequel, in practive it is sufficient to know penalties with very low precision. The order of magnitude of penalties (with cost of an average task as the cost unit) can be expected to be known in most cases.

Risk of getting an error can be estimated if some statistics of previous calculations is available. In the simplest case there are only the error probability and the probability of the correct answer. They are usually known for an algorithm. More subtle estimations are often available.

A priori distribution of possible answers can be also available if there is any statistics. To use virtual screening as an example, there are data bases that give binding energy of ligands and proteins; they allow to estimate probability distribution of different energies. Molecules have different chances to show high predicted binding affinity. These chances are expected to be higher for molecules close by topology to a known ligand [14]. In contrast, molecules with very large number of atoms are less likely to bind well [15]. So, knowledge about chemical structure of molecules can provide estimations of probabilities of the binding energy, i.e., of the possible answers.

History of the search allows to improve these estimations. Some ligands are similar, so results for one serve as source of estimations for another. The same is true for costs C_i of the tasks. In the simplest case they all equal 1; if any additional

information is available, C_i can be estimated more precisely.

Again, results are rather stable with respect to distribution of the answers. What is practically important, is whether one answer is much more rare than the other, or if ratios of probabilities of answers differ much from that of penalties if these answer were wrong. Such facts can be expected to be known, at least in case of simple structure of the answer set, like yes/no or good/normal/bad.

Considering the most general construction is useful also for another reason. Understanding the mechanism helps to organize calculation in an effective way even without precise knowledge about the input data and, then, without exact solution.

3.2 Replication

To protect ourselves from additional losses we need to minimize our total cost. In this paper we analyze replication as a means for that. Too high redundancy does not decrease risk any more, but increases additional work linearly. However, low quorum means high risk and thus expected penalties are too large.

We believe an answer x_i to be correct if it arrives exactly ν_i times from computing nodes; note that the same task can produce other answers on other nodes: we accept the answer already obtained $\nu_i - 1$ times as soon as it is obtained once more, no matter how many times other answers arrived (provided that less than their quorums). In the simplest case of yes or no answers we can distinguish them checking one more carefully [13].

So the total cost of solving a task consists of solving it a few times with the same answer, possibly solving it more times with other answers, and the penalty F_{ij}. This cost is a random variable; we want to minimize its mean value.

To evaluate the mean we need to consider all possibilities of believing a given x_i while some x_j is the correct answer. The final obtained result must be, obviously, x_i: it stops the process completing the quorum. Before that we can have all x_m, in any order, but each can arrive at most $\nu_m - 1$ times (otherwise it would have been believed). The number of all possibilities equals $K_i = \prod_{m=1,m\neq i}^{I} \nu_m - 1$. Having a possibility with a non-negative number $k < K$, we can produce all the numbers $R_{k,m}$ of times an answer x_m has been obtained, $m \neq i$, using the following formulae: $M_{k,0} = k\nu_i$,

$$M_{k,m} = \left[\frac{M_{k,m-1}}{\nu_m}\right], \quad R_{k,m} = \mathrm{mod}(M_{k,m-1}, \nu_m).$$

Note that $R_{k,m}$ depend on i, though the notation hides this dependence. If a k is chosen, the probability that all x_m, $m \neq i$ were obtained exactly $R_{k,m}$ times in any order, the answer x_i was obtained exactly $\nu_i - 1$ times also arbitrarily distributed among other answers, and that it was obtained once more as the final answer is

$$P_{k,i,j} = \frac{\left(\nu_i - 1 + \sum_{m=1,m\neq i}^{I} R_{k,m}\right)!}{(\nu_i - 1)! \prod_{m=1,m\neq i}^{I} R_{k,m}!} \prod_{m=1,m\neq i}^{I} p_{mj}^{R_{k,m}} p_{ij}^{\nu_i}.$$

This is the conditional probability assuming that x_j is truly correct while x_i is believed to be correct, and k-th distribution of other answers has happened. The cost in this case consists of computational expenses for evaluating all rejected answers:

$$\sum_{m=1,m\neq i}^{I} R_{k,m} C_m,$$

expenses on evaluating the accepted task x_i with ν_i replicas: $C_i\nu_i$, and penalty F_{ij}. The sum

$$E_{i,j} = \sum_{k=0}^{\prod_{m=1,m\neq i}^{I} \nu_m - 1} P_{k,i,j} \cdot \Big(\sum_{m=1,m\neq i}^{I} R_{k,m} C_m + C_i\nu_i + F_{ij} \Big)$$

is the expected cost in case x_j is correct while x_i has been accepted. Summing them all up over $i = 1$ to I we get E_j: the expected cost in case x_j is the correct answer. Finally we need to take the *a priori* distribution of correct answers into account to get the expected cost

$$E = \sum_{j=1}^{I} \alpha_j E_j.$$

4. OPTIMIZATION

For given penalties F_{ij} the problem of finding the optimal quorums is to choose such ν_i that E has the minimal possible value. We can assume that $F_{i,i} = 0$ for all i or at least these values are much less compared to other F_{ij}: no penalties for the correct answer. Also error probability must be low enough, otherwise computing has no sense at all: $p_{ii} > 0.5$. Under these assumptions it is clear that if all $\nu_i > \bar{\nu}$ for some $\bar{\nu}$, then the expected penalty is low enough, so that asymptotically $E \sim \sum_{j=1}^{I} \alpha_j C_j \nu_j$ and therefore grows at least linearly with respect to ν. Therefore, choosing high enough ν and then decreasing it, we quickly find an upper bound for all quorums; then further individual reducing of ν_i provides lower bounds.

4.1 Simple cases

In the simplest case there are only two possible answers. Then $I = 2$, $R_{k,m} = k$, $P_{k,i,j} = \binom{\nu_i - 1 + k}{k} p_{3-i,j}^{k} p_{ij}^{\nu_i}$, and

$$E = \sum_{i,j=1}^{2} \alpha_j \sum_{k=0}^{\nu_w - 1} \binom{\nu_i - 1 + k}{k} p_{w,j}^{k} p_{ij}^{\nu_i} \Big(kC_w + C_i\nu_i + F_{ij} \Big)$$

(here $w = 3 - i$). Further simplification is assuming that there is just the correct and the wrong answer, so p_{ij} and F_{ij} are symmetrical 2×2 matrices, $p_{ii} = q$, $F_{ii} = 0$, $\nu_i = \nu$, $C_i = C$. Then α_j do not matter and

$$E = \sum_{i=1}^{2} \sum_{k=0}^{\nu-1} \binom{\nu - 1 + k}{k} p_{3-i,1}^{k} p_{i,1}^{\nu} \Big(kC + C\nu + F_{i1} \Big).$$

Another simple case appears when an answer is absolutely reliable, so that $p_{jj} = 1$ for some fixed j; then, obviously, $p_{ij} = 0$ for all $i \neq j$.

Let us consider the simplest case of two possible answers of equal value with probabilities $q > 0.5$ and $p = 1 - q$ and penalty F in case of the wrong answer has been accepted. Then quorum ν is not optimal if

$$\Delta E = E(\nu + 1) - E(\nu) = \Delta E_0 - AF < 0.$$

Here A is some quantity dependent on p and ν but not on F, while ΔE_0 is ΔE in case of no penalty ($F = 0$) and therefore

is positive. As for A, it is equal to

$$A = p^\nu q^\nu \binom{2\nu-1}{\nu-1}(1-2p).$$

We omit the proof here.

The critical penalty F such that $\Delta E \approx 0$ grows quickly as $p \to 0.5$ and much more quickly as $p \to 0$. For example, in case of $p = 10^{-3}$ we have $A \approx 1.5 \cdot 10^{-8}$ and $\Delta E_0 \approx 1$, so that the critical penalty would be $F^* \approx 0.66 \cdot 10^8$. For $p \approx 10^{-2}$ we have $F^* \approx 0.66 \cdot 10^5$. This asymptotic analysis is applied to the more general case in the next section.

4.2 Asymptotic analysis

Let us assume that probability of an error is negligibly low, so that only threat of penalty justifies taking it into account. Thus we neglect p_{ij} for $i \neq j$ and $1 - p_{ii}$ compared to p_{ii} and 1. Then either the correct answer x_j is obtained ν_j times in a row (other possibilities are too unlikely to consider), or a wrong x_i is accepted with the correct x_j seen any number of times from 0 to ν_{i-1}: all these cases are approximately equally probable. Then $P_{k,i,j}$ becomes simpler: x_i is accepted, another x_j is correct, it is obtained k times with $p_{ij} \approx 1$, $R_{k,m} = k$, m does not vary from 1 to I, instead $m = j$. So, finally:

$$P_{k,i,j} = \binom{\nu_i - 1 + k}{k} p_{ij}^{\nu_i}.$$

If the accepted answer is correct, $P_{k,j,j} \approx 1$, k can be only 0: receiving other answers is highly unlikely. Then

$$E = \sum_{j=1}^{I} \alpha_j \Big(C_j \nu_j + F_{jj} \Big) +$$
$$+ \sum_{j=1}^{I} \alpha_j \sum_{i=1, i\neq j}^{I} \sum_{k=0}^{\nu_j - 1} \binom{\nu_i - 1 + k}{k} p_{ij}^{\nu_i} \Big(k C_j + C_i \nu_i + F_{ij} \Big).$$

We are interested in increment of this cost when a quorum is changed:

$$\Delta E_J = E(\nu_J + 1) - E(\nu_J) =$$
$$\alpha_J C_J + \alpha_J \sum_{i=1, i\neq J}^{I} \binom{\nu_i - 1 + \nu_J}{\nu_J} p_{iJ}^{\nu_i} \Big(\nu_J C_J + C_i \nu_i + F_{iJ} \Big) -$$
$$\sum_{j=1, j\neq J}^{I} \alpha_j \sum_{k=0}^{\nu_j - 1} \binom{\nu_J - 1 + k}{k} p_{Jj}^{\nu_J} \Big(k C_j + C_J \nu_J + F_{Jj} \Big).$$

Now we see that E grows linearly with gradient $\alpha_J C_J$ with respect to ν_J provided that all ν_i are high enough. This means that rare valuable answers, with low probability α_J can be examined carefully without significant losses: ν_J can be taken much more than the optimal value.

Another fact is that not penalties but quantities $p_{ij}^{\nu_i} F_{ij}$ matter. As p_{ij} are low, changes of ν_i modify such quantities very significantly provided that penalties are large compared to costs C_i. Let us say that a quorum $\nu_J > 1$ is equilibrium if $\Delta E_J \geq 0$ while $\Delta E_{J-1} \leq 0$. If a quorum ν_J is equilibrium, it is stable with respect to small changes of the penalty $F_{J,i}$: significant changes of $F_{J,i}$ are such that $F_{J,i} p_{Ji}^{\nu_J}$ remain approximately the same. This means, again, that in case of low risk and high penalties we do not need to know precise values of penalties for accepting a wrong answer: it suffices to know the order of magnitude with precision about p^{-1} where p is the maximal error probability.

A consequence of this is the notion of critical penalties. Let us consider the problem of choosing penalties F_{ij} such that the desired quorums (and thus the desired probability of an error) were optimal. This is a linear optimization problem with an utility function E and constraints of the form $\Delta E_{J-1} \leq 0$. As the utility function has positive coefficients and constraints of the form $F_{ij} \geq 0$ are valid, the problem has a solution provided that at least one admissible point exists. An admissible point is such set of penalties that justifies passing from a quorum ν_{i-1} to ν_i. It is possible that there is no such point, for example if all $p_{Jj} = 0$. Then no penalty is able to make any replication pay. However, if the third term in the expression for ΔE_J is not zero and quorums are high enough, an admissible point always exists and so does a solution to the linear optimization problem for the optimal penalties. These penalties are called critical. If real penalties are close to these ones, even slight difference in computing cost is able to change optimal quorums.

Let us consider another asymptotic case. Assume that the number of possible answers I is so high that probability of receiving a wrong answer twice is negligibly small. Then replication $\nu = 2$ is always sufficient. However, multiple wrong answer are able to arrive prior to the second correct answer, so that amount of work can be high. Let us assume that error risk is the same for all wrong answers, probability of the correct answer is q, $p = 1 - q$ is risk of getting a wrong answer, and cost of each answer is unit. Then we can get from $0 \leq T \leq I - 1$ wrong answers, at most once each, exactly one correct answer at some position and, finally, the second correct answer. Amount of work for each case is $T + 2$, so the mean cost then is

$$E_2 = \sum_{T=0}^{I-1} (T+2)(T+1) p^T q^2 = \frac{2}{q}.$$

The factor $T + 1$ counts number of positions where the first correct answer can be among T wrong ones. So, for reliable computers expected cost is near 2 while for bad ones it can be very high.

We need to compare it with the no-replication case, where the first obtained answer is accepted with penalty F in case of an error. The expected cost is

$$E_1 = 1 + pF.$$

So the critical penalty is

$$F^* = \frac{2 - q}{pq}$$

Again for reliable computers (with small p) critical penalties are of order p^{-1}. Though the minimal possible value is only ≈ 5.83 at $q = 2 - \sqrt{2} \approx 0.586$.

4.3 Desired risk level

Instead of choosing the desired quorums and looking for the critical penalties, one can consider another problem. Choose the desired risk level ρ and demand that the total probability of accepting a wrong answer is at most ρ. This probability is, obviously, E in the special case of $C_i = 0$, $F_{ii} = 0$, $F_{ij} = 1$ for $i \neq j$. Denote this value P. Then the optimization problem has one constraint and looks as

$$E \to \min, \quad P \leq \rho.$$

This problem has a solution provided that $p_{jj} > p_{ij}$ for all i and j: probability of accepting a particular wrong answer would reduce more quickly than that of the right answer. Thus very high quorums make the risk arbitrary low, so that quorums that provide $P \leq \rho$ exist. Then note that E grows with respect to each ν_j provided that all ν_i are high enough and that it grows with respect to each F_{ij}. This fact, together with positiveness of ν_i and $F_{i,j}$, $i \neq j$, guarantees existence of a solution to the optimization problem.

4.4 Different nodes

Assume that we have two groups of computing nodes of different reliability p_{ij} and costs C_i. If we can choose what group to use, we need to compare average overall cost for these two groups under optimal replication. Let us assume that quorums are high enough so that effective expected penalties $p_{ij}^{\nu_i} F_{ij}$ are comparable with costs C_i. Then the quantities $W_{ij} = C_i \nu_i + p_{ij}^{\nu_i} F_{ij}$ can be used to decide which group is better. As an example, let us consider one group with $C_i = 2$ and $p_{ij} = 0.01$ and another group with $C_i = 1$, $p_{ij} = 0.02$. The optimal quorums are $\nu_i = 5$ and $\nu_i = 6$. Then the first group has $W_{ij}^1 = 10 + 0.01^5 F_{ij}$, while the other's $W_{ij}^2 = 6 + 0.02^6 F_{ij} = 6 + 0.64 \cdot 0.01^5 F_{ij} < W_{ij}^1$. This simple example shows that well-chosen replication is able to cope with lower reliability.

5. CONCLUSIONS AND FUTURE WORK

We have considered a model of grid computing with risk of error and threat of penalty and the problem of minimizing the total mean cost of computation consisting of cost of redundant solution of tasks and estimated penalties. The main results are:

- The model, as general as possible; a way to solve the optimization problem.
- The notion of critical penalties, their dependence on risk probabilities.
- Stability of the optimal solution with respect to penalty values, practically important.
- Low harm of overestimated quorums.
- Possibility of improving poor computing by replication.

In future we plan to consider productivity of the Desktop grid together with optimizing the mean cost. Optimal scheduling of replicas is one interesting question. For example, for quorum 2 sometimes it is better to send two replicas hoping for identical answers, while in other cases three replicas should be sent at once so that decision be taken during one time unit. Fighting with saboteurs is even able to force one-by-one scheduling of replicas. We only slightly have considered heterogeneity of computing nodes; though another question is optimal scheduling of optimal number of replicas among heterogeneous computing nodes. The third question to be studied is considering productivity in terms of award that is able to reduce the total average cost. This approach seems to be almost unstudied, so far.

6. ACKNOWLEDGMENTS

The work has been supported by Russian Foundation for Basic Research: projects 16-07-00622 and 15-29-07974.

7. REFERENCES

[1] E. Ivashko, Enterprise desktop grids, in: Proceedings of the II International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC:FAST 2015), Vol. Vol-1502, CEUR Workshop proceedings, 2015, pp. 16–21.

[2] J. Kovács, P. Kacsuk, A. Lomaka, Using a private desktop grid system for accelerating drug discovery, Future Generation Computer Systems 27 (2011) 657–666.

[3] D. P. Anderson, BOINC: A system for public-resource computing and storage, in: Grid Computing, 2004. Proceedings. V IEEE/ACM International Workshop on, 2004, pp. 4–10.

[4] S. Siva Sathya, K. Syam Babu, Survey of fault tolerant techniques for grid, Computer Science Review 4 (2) (2010) 101–120.

[5] D. Kondo, A. A. Chien, H. Casanova, Scheduling Task Parallel Applications for Rapid Turnaround on Enterprise Desktop Grids, Journal of Grid Computing 5 (4) (2007) 379–405.

[6] E. C. Xavier, R. R. S. Peixoto, J. L. M. da Silveira, Scheduling with task replication on desktop grids: theoretical and experimental analysis, Journal of Combinatorial Optimization 30 (3) (2015) 520–544.

[7] A. Benoit, Y. Robert, A. L. Rosenberg, F. Vivien, Static strategies for worksharing with unrecoverable interruptions, Theory Comput Syst 53 (2013) 386–423.

[8] M. K. Khan et al,A Group Based Replication Mechanism to Reduce the Wastage of Processing Cycles in Volunteer Computing, Wireless Personal Communications 76 (3) (2014) 591–601.

[9] E. M. Heien, D. P. Anderson, K. Hagihara, Computing Low Latency Batches with Unreliable Workers in Volunteer Computing Environments, Journal of Grid Computing 7 (4) (2009) 501–518.

[10] M. Taufer, D. Anderson, P. Cicotti, C. L. Brooks III, Homogeneous redundancy: a technique to ensure integrity of molecular simulation results using public computing, in: Parallel and Distributed Processing Symposium, 2005. Proceedings. 2005, pp. 119a–119a.

[11] L. F. G. Sarmenta, Sabotage-tolerance mechanisms for volunteer computing systems (2002).

[12] D. Kondo et al, Characterizing result errors in internet desktop grids, in: A.-M. Kermarrec, L. Bougé, T. Priol (Eds.), Parallel processing. Proceedings of XIII International Euro-Par Conference, Vol. 4641 of Lecture Notes in Computer science, Springer, 2007, pp. 361–371.

[13] I. Chernov, N. Nikitina, Virtual screening in a desktop grid: Replication and the optimal quorum, in: Parallel Computing Technologies: 13 International Conference, Springer, 2015, pp. 258–267.

[14] D.E. Patterson et al., Neighborhood behavior: a useful concept for validation of "molecular diversity" descriptors, Journal of Medicinal Chemistry 39 (1996) 3049–3059.

[15] M.M. Hann, A.R. Leach, G. Harper, Molecular complexity and its impact on the probability of finding leads for drug discovery, Journal of Chemical Information and Computer Sciences 41 (2001) 856–864.