Title	Influence of performance prediction inaccuracy on task scheduling in grid environment
Author(s)	Zhang, Yuanyuan; Inoguchi, Yasushi
Citation	Lecture Notes in Computer Science, 3399: 838-844
Issue Date	2005
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/3973
Rights	This is the author-created version of Springer, Zhang, Yuanyuan ; Inoguchi, Yasushi, Lecture Notes in Computer Science, 3399, 2005, 838-844. The original publication is available at www.springerlink.com. http://springerlink.metapress.com/content/u7rg2k9am2efhynt/?p=e738f66809704533871cb33e9bce26d7&pi=0
Description	

Influence of Performance Prediction Inaccuracy on Task Scheduling in Grid Environment

Yuanyuan Zhang[1] and Yasushi Inoguchi[2]

[1] Graduate School of Information Science,
[2] Center for Information Science,
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Tatsunokuchi, Ishikawa, 923-1292, Japan
{yuanyuan,inoguchi}@jaist.ac.jp ⋆

Abstract. *In this paper, we study the influence of performance prediction inaccuracy on task scheduling in grid environment from the context of task selection and processor selection, which are two critical phases in task scheduling. Formulas are established for the degree of misprediction, the probability that the predicted values for the performance of tasks and processors reveal different ordering characteristics from their real values. The impacts of different parameters on the degree of misprediction are also investigated extensively. Evaluation results show that an underestimate of performance can result in greater influence on task scheduling compared with an overestimate, while higher heterogeneity results in smaller influence.*

Keywords: grid computing, task scheduling, performance prediction, task selection, processor selection

1 Introduction

Grid computing[1] is becoming increasingly popular recently. Task scheduling, the problem of scheduling tasks to processors so that all the tasks can finish their execution in the minimal time, is a critical component for achieving high performance in grid environment. Usually the scheduling process of a task scheduling algorithm involves two phases: task selection and processor selection. In task selection phase, the tasks are sorted in a list according to some criterion related with the workloads of the tasks; while in processor selection phase, the first task in the list is allocated to a processor based on another criterion typically related with the speeds of the processors. Therefore, the prediction for the task workloads and processor speeds is critical for achieving satisfying performance in grid task scheduling system. Usually such prediction are executed by some performance prediction tools such as NWS[2].

⋆ This research is conducted as a program for the "21st Century COE Program" by Ministry of Education, Culture, Sports, Science and Technology, Japan.

Existing scheduling algorithms typically assumed that the task scheduler has perfect knowledge about the performance of both tasks and processors. However, although nowadays the performance prediction tools can provide increasingly accurate prediction, it is still impossible to achieve absolutely accurate prediction since grid is a highly dynamic environment[1]. Therefore the performance of task scheduling algorithms will be influenced by such inaccurate prediction, and different task scheduling algorithms reveal different degrees of sensitivity to the inaccurate prediction. In this paper, we would focus on the study of the influence of performance prediction inaccuracy on task scheduling from the perspectives of task selection and processor selection in grid environment.

The remainder of this paper is organized as follows: in next section, we analyze the influence of prediction inaccuracy on task scheduling, introduce the concept of degree of misprediction, and establish related formulas. The impact of the parameters in the formulas for the degree of misprediction is evaluated in section 3. Finally, the paper is concluded in section 4.

2 Analysis of the Influence of Prediction Error on Task Scheduling

2.1 Task Selection

Grid computing is a highly heterogeneous and dynamic environment. Task scheduling problem in such a heterogeneous and dynamic environment is much more difficult than that in homogeneous system. Furthermore, since it is impossible to obtain absolutely accurate prediction for the workloads of tasks because of the dynamicity, the actual workloads of tasks will be different from the predicted values, and this will influence the performance of task scheduling algorithms. For example, if the actual workload of task T_i is less than that of task T_j, while because of prediction error, the predicted value of T_i is more than that of T_j, then we will make wrong scheduling decision if we schedule the tasks based on their workloads.

In this paper we focus on the study of a grid application which is composed of a set of independent tasks. The actual workloads of these tasks are independent identical distribution(i.i.d.) random variables. In grid scheduling system when performance prediction tools are used to predict the performance of the tasks, the predicted errors usually lie in an interval of the actual workloads according to some probability distribution.

Suppose T_1 and T_2 are two tasks in a grid application and their actual workloads are denoted by positive numbers x_1 and x_2 respectively. The prediction errors of T_1 and T_2, y_1 and y_2, are independent random variables and follow some probability distribution in the ranges of [-ax_1,bx_1] and [-ax_2,bx_2], where the possible value fields of a and b are [0,1) and [0,∞) respectively. The probability density function of prediction error is denoted by $g(y)$. For the predicted workloads of T_1 and T_2, denoted by z_1 and z_2, we have the following equations:

$$z_1 = x_1 + y_1; \qquad z_2 = x_2 + y_2. \tag{1}$$

Usually the task scheduling algorithms schedule tasks based on their predicted workloads, for example, schedule the task with the largest workload first or with the smallest workload first, so the prediction inaccuracy has remarkable influence on the performance of scheduling algorithms when the actual workload of T_1 is smaller than that of T_2 while because of the prediction errors, the predicted value of T_1 is greater than that of T_2. We call such situation *misprediction.* Because different performance prediction tools have different degrees of prediction inaccuracy, they can arouse different degrees of misprediction. Therefore, what we are interested in is the probability of the misprediction to happen, i.e., $P(z_1 > z_2|x_1 < x_2)$, which is called *degree of misprediction for two tasks* here.

The above probability can be converted into:

$$P(z_1 > z_2 \mid x_1 < x_2) = P(y_1 > y_2 + x_2 \text{ - } x_1|x_1 < x_2) \qquad (2)$$

where $y_1 \in$ [-ax_1, bx_1], and $y_2 \in$ [-ax_2,bx_2].

In the coordinate system of y_1 and y_2, the inequality $y_1 > y_2 + x_2$ - x_1 is the area above the line L: $y_1 = y_2 + x_2$ - x_1, and the probability of P($y_1 > y_2 + x_2$ - $x_1|x_1 < x_2$) can be expressed by the area of the overlapping region between L and the rectangle surrounded by the lines y_1 = -ax_1, y_1 = bx_1, y_2 = -ax_2 and y_2 = bx_2 in the y_1, y_2 coordinate system.

About the overlapping region, we have the following conclusion:

There are only two cases for the overlapping between L and the rectangle: they either don't overlap or overlap in a triangle region.

Proof. Case(1): This case is shown in figure 1. The line which is parallel to L and passes the point (bx_1,-ax_2) is $y_1 = y_2$ + ax_2 + bx_1. If L is above this line, then we can see intuitionally that there is no overlapping between L and the rectangle. That is to say, if x_2 - $x_1 \geq$ ax_2 + bx_1, i.e., if (1-a)$x_2 \geq$ (1+b)x_1, the probability P($y_1 > y_2 + x_2$ - $x_1|x_1 < x_2$) equals to 0. It can be expressed mathematically as:

$$P(y_1 > y_2 + x_2 - x_1|x_1 < x_2) = 0 \qquad \text{if } (1\text{-a})x_2 \geq (1 + b)x_1, x_2 > x_1.$$

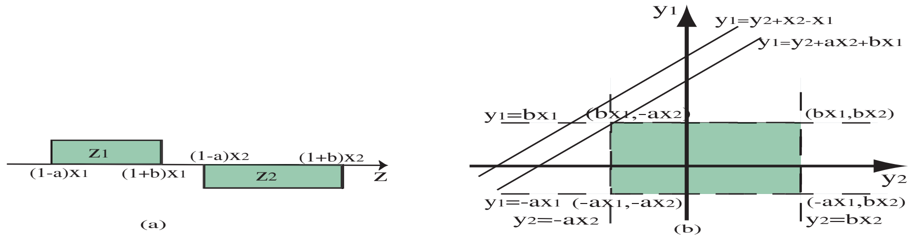

Fig. 1. The case when there is no misperception. (a)the value fields of predicted workloads which don't overlap; (b)the situation of the degree of misperception.

Case(2): If L is under the line $y_1 = y_2$ + ax_2 + bx_1, then it will overlap with the rectangle. The probability for the misprediction to happen equals to the area of the overlapping region. The line which is parallel to L and passes the point (-ax_1,-ax_2) is: $y_1 = y_2$ + a(x_2 - x_1). Since $0 \leq$ a < 1, a(x_2 - x_1) $< x_2$ - x_1. So L must be above the line $y_1 = y_2$ + a(x_2 - x_1) in any way. From figure 2 we can see that the overlapping region can only be a triangle.

So the proof is completed.

In the case L and the rectangle overlaps in a triangle, the area of the overlapping region, i.e., the probability P($y_1 > y_2 + x_2$ - $x_1|x_1 < x_2$), can be expressed by the double integral of the probability density functions g(y_1) and g(y_2). So we have the following equation:

$$P(y_1 > y_2 + x_2 - x_1 \mid y_1 < y_2) = \int_{-ax_2}^{(1+b)x_1-x_2} \int_{y_2+x_2-x_1}^{bx_1} g(y_1)g(y_2)dy_1dy_2$$

if (1-a)x_2 < (1+b)x_1.

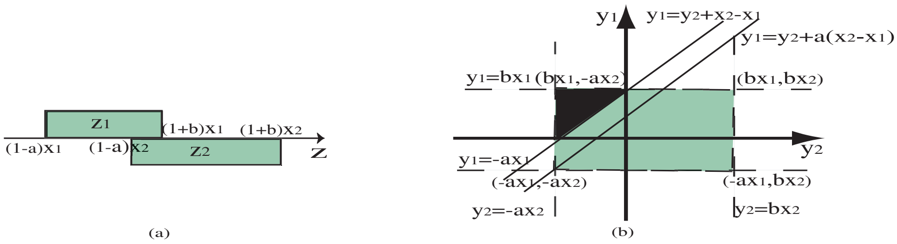

Fig. 2. The case when misperception happens. (a)the value fields of the predicted workloads which overlap; (b)the situation of the degree of misperception.

If the probability density function of the actual workload of a task in the grid application is f(x) and the value field of the actual workload x is [xl,xu], then the *degree of misprediction for the grid application,* which is denoted by *DM* , is defined as the average of the degree of misprediction between any two tasks in the application. In virtue of the equation for the degree of misprediction between two tasks as shown before, *DM* can be expressed as:

$$DM = \int_{xl}^{xu} \int_{xl}^{\frac{1+b}{1-a}x_1} \int_{-ax_2}^{(1+b)x_1-x_2} \int_{y_2+x_2-x_1}^{bx_1} f(x_1)f(x_2)g(y_1)g(y_2)dy_1dy_2dx_2dx_1$$

This result is independent with the specific function for *f(x)* and *g(y)*.

2.2 Processor Selection

In processor selection phase, usually a task scheduling algorithm selects a processor according to the predicted computational speed of the processors, for example, select a fastest processor or a slowest processor, while because grid is a highly dynamic environment, the performance prediction tools usually can not provide entirely accurate prediction for the processor speeds, and such prediction inaccuracy will affect the performance of the task scheduling algorithm.

Suppose there are m heterogeneous processors in the grid system. The actual computational speed of processor P_i(1≤ i ≤m) is denoted by s_i. The prediction error of s_i, which is denoted as t_i, typically lies in a range of s_i following some probability distribution. The prediction errors of different processors are independent random variables. Let s_i and s_j be the actual speeds of processor P_i and P_j respectively. The prediction errors for them are t_i and t_j. t_i and t_j are independent random variables. The value fields of t_i and t_j are respectively [-as_i,bs_i] and [-as_j,bs_j] with the probability density function h(t), where $0 \le a < 1, b \ge 0$(a and b here are different from that in the above subsection). The predicted speeds of P_i and P_j are w_i and w_j. We have the following equations:

$$w_i = s_i + t_i; \qquad\qquad w_j = s_j + t_j. \tag{3}$$

The degree of misprediction for two processors is defined as the probability of the event that the actual computational speed s_i is smaller than s_j, while because of prediction errors, the predicted speed w_i is greater than w_j, that is, the probability $\mathrm{P}(w_i > w_j | s_i < s_j)$. This can be further transformed to:

$$\mathrm{P}(w_i > w_j | s_i < s_j) = \mathrm{P}(t_i > t_j + s_j - s_i | s_i < s_j). \tag{4}$$

Following a similar way as that in task selection, we can derive the equation for processor selection:

$$P(t_i > t_j + s_j - s_i | s_i < s_j) = \begin{cases} 0 & if (1-a)s_j \geq (1+b)s_i \\ \int_{-as_j}^{(1+b)s_i - s_j} \int_{t_j + s_j - s_i}^{bs_i} h(t_i)h(t_j)dt_i dt_j & else \end{cases}$$

The degree of misprediction for the scheduling of a task in a grid system with m processors, which is denoted by DM_P**,** is defined as the average of the degree of misprediction between any two processors. In virtue of the equation for the degree of misprediction between two processors as shown before, DM_P can be expressed as:

$$DM_P = \frac{1}{C_m^2} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} P(t_i > t_j + s_j - s_i | s_i < s_j). \tag{5}$$

3 Study of Evaluation Results

We present the results from our evaluations which assess the impact of the parameters in formula (5), the influence of prediction inaccuracy on processor selection. The results are shown in figures 3(a)-(f), where the horizontal axis in every figure is the combination of the values of a and b, and the vertical axis is DM_p, the degree of misprediction for processor selection.

The parameters in formula (5) are a, b, m, s_i($1 \leq$ i $\leq$ m) and h(t). Uniform distribution for the prediction error of processor execution speed is assumed, that is, the probability density function of h(t) is assumed to be

$$\mathrm{h(t)} = \frac{1}{(a+b)s},$$

while s is the actual execution speed of a processor. We also assume that there are 6 processors in the grid computing system(m=6) and the actual execution time of the first processor is s_1=500. To evaluate the impact of the grid heterogeneity on the degree of misprediction, two groups of estimations are conducted: in figure 3(a), (c) and (e) the actual execution times of the processors increase successively by the increment of 5(s_{i+1} - s_i = 5, $1 \leq$ i $\leq$ m-1), while in figure 3(b), (d) and (f) the increment is assumed to be 250(s_{i+1} - s_i = 250, $1 \leq$ i $\leq$ m-1).

From the comparison of the graphs on the left with that on the right, we can see that, the degree of misprediction decreases as the heterogeneity of grid computing system increases. In figure 3(a) and (b) the range of (a,b) shifts gradually from (0.1,0.9) to (0.9,0.1), that is, the prediction error shifts gradually from overestimate to underestimate. From figure 3(a) and (b) we can see that the degree of misprediction increases as the prediction inaccuracy changes from overestimate to underestimate, moreover, such increase is fleet in highly heterogeneous grid system, while in grid system with low heterogeneityp, the degree of misprediction increases slowly with the shift of (a,b). In figure 3(c) and (d), the predicted error shifts from [-0.1s_i,1.9s_i] to [-0.9s_i,1.1s_i]. Comparing figure 3(c) with (a) and (d) with (b), we can see that the degree of prediction increases as the range of prediction error increases, and higher heterogeneity results in faster increase. In figure 3(e) and (f), (a,b) increases from (0.1,0.1) to (0.9,0.9), while the average prediction error remains constantly to be 0. In figure 3(e) and (f), the degree of misprediction increases as the range of prediction error increases.

Although we only present the results for the case of resource selection here, it is expected that the task selection phase can reveal similar results.

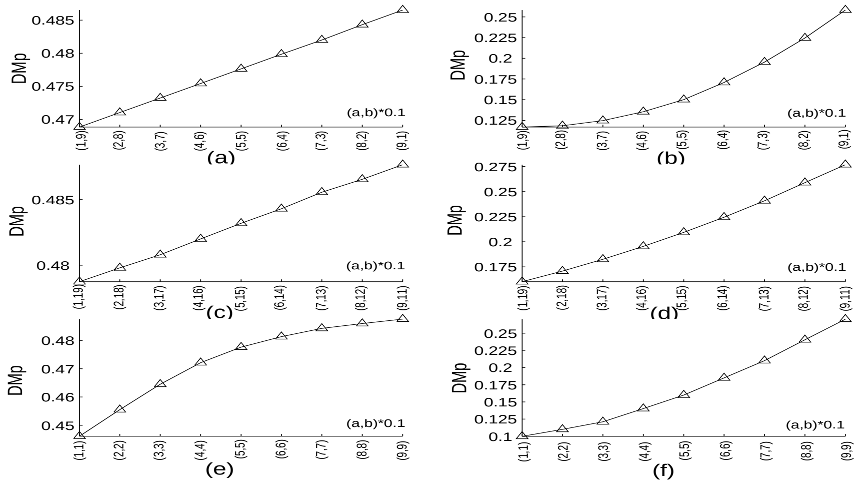

Fig. 3. Influence of parameters a,b and processor heterogeneity on the degree of misprediction. (a),(c),(e): low heterogeneity, (b),(d),(f): high heterogeneity.

4 Conclusion

The prediction inaccuracy for the performance of tasks and processors usually exists so that influences the performance of task scheduling algorithms in grid computing. This paper studies such influence from the perspective of task selection and processor selection. Evaluation results show that an underestimate of performance can result in greater influence on task scheduling compared with an overestimate, while higher heterogeneity results in smaller influence. We hope our results can provide some references for task scheduling in grid environment.

References

1. I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, San Fransisco, CA, 1999.
2. R. Wolski, N. Spring, and J. Hayes, "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing," *Journal of Future Generation Computing Systems*, pp. 757-768, Vol. 15, , Oct., 1999.