International Journal of Innovative
Computing, Information and Control
Volume **6**, Number **9**, September **2010**

ICIC International ©2010 ISSN 1349-4198
pp. 3887–3909

PROBABILISTIC SERVICE PARTITION FOR PARALLEL AND DISTRIBUTED COMPUTING

Kuen-Fang Jea[1] and Jen-Ya Wang[1,2]

[1]Department of Computer Science and Engineering
National Chung-Hsing University
Taichung 40227, Taiwan
kfjea@cs.nchu.edu.tw; jywang@sunrise.hk.edu.tw

[2]Department of Computer Science and Information Management
Hungkuang University
Sha Lu 43302, Taiwan

Received March 2009; revised September 2009

Abstract. *In this paper, we consider an optimization problem that aims to minimize the average waiting time for distributed services with different processing complexities and access probabilities. It is motivated by the fact that there are many large-scale scientific projects and commercial applications (e.g., image processing in astronomy), and their waiting time needs to be lowered down in order to maintain customer satisfaction. We first demonstrate several useful properties of this problem by mapping it to the Euclidean space* $\mathbf{R}^n$*. Utilizing them, we then develop a gradient-based method for dividing and distributing services to multiple machines. The theoretical analyses show that the proposed method converges linearly and the resultant average waiting time is near optimal. Finally, we present experimental results that confirm the convergence speed and solution quality of the proposed method. Using the proposed method, a service provider requires only a little execution time to deploy his/her services on multiple machines and provides users with a near-optimal average waiting time for their service requests. The proposed method can be extended to other similar optimization problems (e.g., vehicle routing problem) and promisingly achieves the same near-optimal results.*
Keywords: Optimization, Gradient, Distributed computing, Parallel computing, Average waiting time

1. **Introduction.** Today's computing is increasingly data-intensive or time-consuming and it should be partitioned into many services and allocated among multiple machines. It is commonly found in practice that parallel computing or grid computing succeeds in supporting large-scale projects or applications. This can be seen in a variety of scientific projects; several concrete examples of which include: protein folding [45], image processing in astronomy [42], satellite data archiving [26], and physics data sharing [11]. Likewise, many types of commercial applications requiring intensive computing are quite often seen today, such as online games [33], Google search [15], and dynamic instantiation [40]. Most of the above examples process petabyte-scale datasets and serve a great many users. Owing to the enormous amount of data and considerable computational complexity, such a project or an application is not suitable for implementation on a single machine; therefore the datasets/services need to be divided and distributed among multiple machines. However, an arbitrary partition will lead to poor performance (e.g., a long waiting time). As a result, many partition methods have been developed to improve the performance for these projects or applications.