Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 461–470

Procedia Computer Science

International Conference on Computational Science, ICCS 2011

Management of Non-functional Attributes of Parallel Components

Yunfeng Peng, Changjun Hu, Chongchong Zhao, Shigang Li, Shucai Yao

Department of Computer Science and Technology,
University of Science and Technology Beijing
Beijing, China

Abstract

In this paper, we present an extension to the CCA component architecture. The extension defined a minimal set of non-functional attributes of parallel components. We have[a] implemented the common CCA components for the management of these attributes. Parallel components can provide some non-functional interfaces optionally. And they provide related information to the common components through these interfaces. For the optimization of parallel components implementations, the component developer should implement the attributes management parts specific to certain parallel components. The tests of the management of six of them show that our management mechanism can improves the performance of parallel components. And the test of management of the other three shows that this is an easy way of controlling parallel component execution. And it is very efficient. A real application test shows the practicability of our proposal.

Key words: parallel component, non-functional attribute, CCA, attribute management, resource management

1. Introduction

Recently, in the area of parallel computing, raising the productivity of parallel computing software is gaining more importance for two reasons. On one hand, the development of large scale parallel applications is still at stage of handwork, causing it very hard to maintain and reuse the software. On the other, new hardware architecture is available, which needs high portability and high productivity of parallel software. Based on common component technologies like COM [1], CORBA [2] and EJB [3], parallel component becomes a technology to solve these problems. Parallel components are components used to compose high performance scientific computing applications. Concerto project [4] introduced a middleware platform for parallel adaptive components. This platform provides a mechanism for resource modeling and monitoring. Furmento Nathalie advanced a layered component architecture for HPC Applications [5]. It separates component abstraction and implementation. And Lei Zhao [6] shows a method for predictive performance modeling of parallel component composition.

Researchers from American national labs and academic institutions founded Common Component Architecture (CCA) forum for parallel components [7]. In this architecture, components written in different languages can interoperate through Scientific Interface Definition Language (SIDL) [8]. CCA gives the least requirements on components [7]. This requirement is a component only has to implement the CCA's Component interface. This interface only defines one method, "setServices( )" [7]. The most important benefit of CCA is that it can incorporate the legacy codes. CCA puts a strong emphasis on performance. It is targeted for high performance scientific computing. Both parallel and distributed computing can be used in CCA [7].

These existing parallel component technologies have changed the way in which scientific computing software was designed. But some attributes of parallel components orthogonal to the scientific function need to be paid more attention. These attributes are somewhere referred as QoS (Quality of Service) [9] [10]. Sometimes, they can affect the performance of parallel components. They may also help the users to control the execution of parallel components. We defined these attributes as non-functional attributes in this paper. These non-functional attributes may appear and be discussed in other software applications [11]. And we believe that they can improve the usefulness of parallel components software.

In this paper, we present an extended component architecture to CCA. This architecture provides a unified management to the

* Yunfeng Peng..Tel.: +8613520368146
E-mail address:peng_ustb@yahoo.com.cn.

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
doi:10.1016/j.procs.2011.04.048

non-functional attributes of CCA functional components. We provide the management of nine attributes of functional components. They are performance, resource requirement, deployment, schedule, load balance, adaptive, execution, transaction, security and access. Our extension gives a convenient way for managing the non-functional attributes of parallel components applications.

The most important contribution is the separation of attributes management in parallel components applications. First, the management of non-functional attributes is treated as a special unit in the parallel components applications. Second, this management is divided into two parts. One part contains some non-functional components. They define the management part shared by all functional components. These components are common components to all parallel application components. The other part includes some optional non-functional interfaces. These interfaces can be provided by the functional components. The methods implemented in these interfaces realize the management parts specific to different functional components. And this management is extensible. The users can implement their own methods for their components. The tests results show that our management is effective. And this management can improve the performance of parallel components applications. Some reflective systems also gave the concept for separating interfaces for meta-level description of components [12]. They divided reflection into two parts. Structural reflection can change the functional aspects of the component [12]. Behavioral reflection is used to dynamically insert some interceptors for application monitoring [13]. Our paper focuses on the non-functional attributes of parallel components. How to control and manage these attributes in a parallel components application is our primarily concern.

The remaining sections of this paper are organized as follows: Section 2 describes the extended component architecture. Section 3 shows the management of platform resources. Section 4 introduces of non-functional attributes management mechanism. Section 5 contains some examples and tests. Section 6 concludes the paper. And In this paper, we use FC to denote functional component, and NFC stands for non-functional component.

2. Extended component architecture

CCA researches play an important role in designing scientific computing software [14]. But for a parallel component, there are many important attributes beyond a scientific function. For example, resource management and load balance can affect the performance of parallel software greatly [15] [16]. These attributes somewhere referred as QoS (Quality of Service) [9] [10], are orthogonal to the scientific functions of parallel components. We defined these attributes as non-functional attributes of parallel component. In a survey conducted by the high performance computing lab of University of Science and Technology Beijing, 96.4% of the 2000 scientific computing software users questioned said they had met the problem beyond the scientific function requirements. We defined a series of non-functional problems in a problem list. Between the users who met the non-functional problems, 93% gave at least one of nine specific problems from the problem list. These nine problems they met are named separately as adaptive, load balance, deployment, performance, schedule, execution, resource requirement, security and transaction in our list. These attributes are especially noteworthy when the execution environments are heterogeneous platforms. And the variation of implementations of parallel components makes the management of their attributes difficult. As CCA is a basic definition for parallel components. It tries to establish a well-accepted standard for scientific computing software components. It wants to incorporate as many components as possible. This made CCA not so mature for the management of non-functional attributes of parallel components.

So we defined the attributes associated with these nine problems as a minimal set of non-functional attributes of parallel components. These nine attributes are defined as follows. We defined adaptive as the capability of a parallel component. With this attribute, component can react to specific event statically or dynamically. This event can be hardware resource change, input data variation or other user defined events. The reaction can be parallel degree variation, data partition variation or other changes to the parallel component. We defined load balance as another capability of a parallel component. It stands for that a parallel component can migrate the load of it from one processor to another. We defined deployment in this way. A parallel component can have a deployment attribute. Then this component can select a suitable resource from a set of hardware resources. Deployment of this component on this resource can have better performance than others. A component with performance attribute can get its running performance data. It also can predict the performance of itself on certain resources. This parallel component can also manage its performance data. Schedule attribute give the parallel component a way to schedule tasks in the component. Base on certain policies, tasks can be scheduled in a user defined sequence. Execution attribute can suspend the execution of a parallel component. It also can continue the execution from the suspend point when the user requires. This attribute can also stop the execution of a component and exit. Resource requirement attribute gives the user a means to set and get the resource requirement of a parallel component. Security attribute controls the secure access of a parallel component. This attribute sets and performs the access authorities of this component. Only allowed users or methods can access permitted parts of the component. Transaction attribute controls the transactional method execution in parallel components. In a transactional method, some transactional operations are performed on some parallel components. If all these operations success, this method will be successful executed. If not, this method failed and will be executed from the beginning.

We divided parallel components into functional components and non-functional components. A functional component (FC) is a component performs some scientific functions the user needed in the original applications. A non-functional component (NFC) is a component we designed for the management of non-functional attributes of functional components. A NFC is a common component to all parallel applications. A CCA component can define “provides” ports and “uses” ports. Here ports are abstract interface. We take ports as interfaces for simplicity. A “provides” port contain the methods a component implements. A “uses”