Available online at www.sciencedirect.com
ScienceDirect
Future Generation Computer Systems 24 (2008) 354–370
FGCS
FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

COHESION — A microkernel based Desktop Grid platform for irregular task-parallel applications

Sven Schulz, Wolfgang Blochinger*, Markus Held, Clemens Dangelmayr

University of Tübingen, Symbolic Computation Group, Sand 14, D-72076 Tübingen, Germany

Received 26 March 2007; received in revised form 13 June 2007; accepted 16 June 2007
Available online 28 June 2007

Abstract

We present COHESION, a novel approach to Desktop Grid Computing. A major design goal of COHESION is to enable advanced parallel programming models and application specific frameworks. We focus on methods for irregularly structured task-parallel problems, which require fully dynamic problem decomposition. COHESION overcomes limitations of classical Desktop Grid platforms by employing peer-to-peer principles and a flexible system architecture based on a microkernel approach. Arbitrary modules can be dynamically loaded to replace default functionality, resulting in a platform that can easily adapt to application specific requirements. We discuss two representative example applications and report on the results of performance experiments that especially consider the high volatility of resources prevailing in a Desktop Grid.

Keywords: Desktop Grids; Grid computing; Software architecture; Peer-to-peer systems

1. Introduction

Grid Computing is increasingly recognized as a major building block of the eScience vision. It provides researchers uncomplicated access to resources beyond the boundary of a single institution. Most importantly, Grid Computing can deliver unprecedented computing power.

In this paper we deal with Peer-to-Peer (P2P) Desktop Grid Computing. This discipline of Grid Computing integrates methods from high performance computing with state-of-the-art concepts from the realm of distributed systems. Desktop Grid systems harness underutilized resources of end-user computers for tackling computationally hard problems. This approach to Grid Computing is gaining momentum, since it is able to deliver considerable computing power at virtually no extra cost. Small scale installations, e.g. comprising the workstations of a department, and also large scale, Internet-wide approaches (also known as Global Computing) have been successfully implemented.

However, Desktop Grids differ significantly from traditional parallel systems. Particularly, resources exhibit a high degree of volatility and heterogeneity: Depending on the usage patterns of the participating desktop computers, resources with considerably differing capabilities join and leave the grid in an unpredictable manner. Delivering sustained computing power in such environments poses enormous challenges to system and application designers.

As a consequence, Desktop Grid applications are most often based on trivial parallelism. The problem at hand is decomposed into independent subproblems, which can be farmed out for computation without further communication among the subproblems. Extending the scope of the Desktop Grid approach towards more non-trivial parallel applications involves aspects on all levels of parallel system design. In this paper, we present our P2P Desktop Grid platform COHESION (available from [1]). Our research aims at laying the system-level foundations for more tightly coupled parallel computations requiring complex interaction patterns among the participating nodes. The main target domain of COHESION are irregular task-parallel applications. Typically, these applications employ fully dynamic problem decomposition based on a distributed task pool execution model. In particular, we make the following contributions:

* Corresponding address: University of Tübingen, WSI/SR, Sand 14, D-72076 Tübingen, Germany. Tel.: +49 7071 29 70469; fax: +49 7071 29 5160.
E-mail address: blochinger@informatik.uni-tuebingen.de (W. Blochinger).

0167-739X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2007.06.005

- We demonstrate, how advanced P2P principles and techniques can be assembled to create a comprehensive collection of core functionality required for parallel computing.
- We show, that a P2P approach to Desktop Grid Computing also has a fundamental impact on the architecture of the system. We present an appropriate solution based on an industrial strength microkernel technology.
- We explain, how sophisticated parallel programming models and application specific parallel frameworks can be realized on top of the abstractions and primitives provided by COHESION.

The remainder of our paper is organized as follows: In Section 2, we give an overview of related work. Section 3 discusses a P2P based execution model for irregular task-parallel applications. In Section 4, we identify requirements for P2P based Desktop Grid Computing. Section 5 gives a comprehensive description of the design, architecture, and implementation of COHESION. In Section 6, we present example applications and discuss the results of performance measurements.

2. Related work

Grid computing has been a highly active area of research in the last decade. The textbook [2] provides a comprehensive overview of all major topics.

One main goal of Grid research has been enabling *virtual organizations* [3]. Within a virtual organization, organizationally owned resources (possibly of considerable value) can be shared among autonomous and often geographically dispersed institutions without sacrificing local authority.

Desktop Grid Computing [4] aims specifically at harnessing unused resources from non-dedicated end-user machines. This (largely orthogonal) discipline of Grid research has been motivated by today's pervasiveness of information technology and the resulting plethora of exploitable computing power [5]. According to the scale and the relationship among resource providers and resource consumers we can distinguish different approaches to Desktop Grid Computing. Computation exchange platforms (e.g. CompuP2P [9]) establish a symmetric relationship between resource providers and resource consumers by creating virtual markets for trading resources. A more asymmetric relationship between these parties is typical for volunteer (global) computing platforms (e.g. BOINC [10]) and for Enterprise Desktop Grids (e.g. Entropia [11]).

Both approaches to Grid Computing ultimately pursue the same goal, aggregating resources beyond local administrative domains. However, they face different requirements and constraints, e.g. target communities (limited trust versus no trust) or nature of resources (high-end versus end-user) [6].

System architectures for building virtual organizations (e.g. Globus toolkit [7]) must specifically deal with interoperability issues, like standardization of protocols and interfaces. It turned out, that these requirements can be effectively met by leveraging industry standards from the realm of web services [8]. However, this approach results in rather complex systems and imposes high organizational requirements, e.g. highly qualified personnel. In contrast, architectures for constructing Desktop Grids must specifically reflect the high degree of resource volatility. Also, only little administrative overhead is acceptable, since typically no additional personnel is available for operating Desktop Grid installations. As a consequence, more lightweight and modular system architectures become mandatory, since they reduce software and runtime complexity and can also adapt to the prevailing dynamism.

Next, we classify several Desktop Grid platforms, according to their basic architectural approach and subsequently, we elaborate on state-of-the-art techniques for building modular distributed software.

2.1. Client/server Desktop Grid platforms

Client/server or multi-tier Desktop Grid platforms employ a proven and well understood operational model. Thus, they have reached a considerable degree of maturity and stability and are particularly suited for commercial or mission critical applications. However, due to their centralized nature, programming models requiring complex interaction patterns among the participating nodes cannot be realized efficiently. Typically, such platforms exclusively support the *bag of tasks* or *master/worker* parallel programming model. As a consequence, client/server platforms are mainly suited for applications, which are based on trivial parallelism or for plain high throughput computing.

Well-known examples of this class of Desktop Grid platforms are BOINC and Entropia. Basically, these platforms are very similar. Differences can be observed concerning the client-side security model (signed code versus native sandboxing), support of heterogeneous environments, and functionality for dynamically integrating applications.

In [12] a prototypical software stack for large scale distributed systems (LSDS) is presented that specifically addresses resource volatility and respective application programming models. XtremWeb is a Desktop Grid platform, which implements a subset of this architecture by a three-tier approach. It employs a coordinator for connecting clients and workers, which is (currently) implemented in a centralized way.

2.2. P2P Desktop Grid platforms

In recent years, several (experimental) platforms for high performance computing based on P2P principles have been described in the literature. Subsequently, we discuss some prominent representatives.

The JNGI [13] system is mainly targeted towards coarse-grained, embarrassingly parallel applications. It supports the master/worker programming model, thus being very similar to client/server platforms at the programming model level. However, the JNGI architecture focuses on extreme scalability and reliability. It pursues a self-organizing approach, taking advantage of the capabilities of the JXTA P2P platform. JNGI's