Research Reports in Computer Science

Band 5

Gérald Jean-Pierre Prétôt

HPCS: Client-Server Support for High-Performance Computing

Shaker Verlag
Aachen 1999

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Prétôt, Gérald Jean-Pierre:
HPCS: Client-Server Support for High-Performance Computing /
Gérald Jean-Pierre Prétôt. - Als Ms. gedr. - Aachen : Shaker, 1999
(Research Reports in Computer Science ; Bd. 5)
Zugl.: Basel, Univ., Diss., 1999
ISBN 3-8265-6700-5

Printed in Germany.

ISBN 3-8265-6700-5
ISSN 1436-6967

Shaker Verlag GmbH • P.O. Box 1290 • D-52013 Aachen
Telefon: 0049/2407/9596 - 0 • Telefax: 0049/2407/9596 - 9
Internet: www.shaker.de • eMail: info@shaker.de

Abstract

Distributed and parallel computing are well-established and successful fields. In both certain levels of standardization have been achieved and reliable tools are available for software development and deployment. However, so far hardly any effort has been made, to *integrate* the two worlds in a generic way. This is the goal of the work presented here.

This thesis introduces a new model for *client-server support for high-performance computing* called *HPCS*. The model offers a generic approach to making parallel, high-performance implementations of *compute services* accessible to sequential clients in the form of software *components*. It exploits the abstraction provided by *parallel data structures* and *parallel procedures* to allow *transparent* access from remote clients. In HPCS parallel data structures are presented as *abstract data objects* and parallel procedures as *services*. Access is provided using standardized client-server *interfaces* such as the ones specified by the *CORBA* standard. This is done in a way that does not require the service programmer to be concerned with the client-server interface nor the client programmer with the parallel nature of the service implementation.

Call-persistent data objects and a minimal set of methods for their manipulation are proposed as a mechanism to work with parallel data structures that remain on the server between different service calls. A *parallel request handler* split into two components, the *Multiplexer* and the *Dispatcher*, provides transparent access to parallel services.

ACS, a concrete implementation of HPCS targeting the parallel programming language *ALWAN* and *CORBA* implementations is presented. Thus, ACS enables writing high-performance services in ALWAN which are translated into executable parallel programs and interfaces that allow access to the services from CORBA clients.

Sources of *overhead* are discussed and some measurements are presented that provide a feeling for the possible behavior of an HPCS application. Examples from the fields of *volume rendering*, *signal processing* and *computational fluid dynamics* demonstrate how ACS is used in applications.