The Ibis Project: Simplifying Grid Programming and Deployment

Henri Bal Jason Maassen Rob van Nieuwpoort Thilo Kielmann
Niels Drost Ceriel Jacobs Frank Seinstra Roelof Kemp Kees Verstoep
Department of Computer Science, VU University, Amsterdam, The Netherlands
{bal, jason, rob, kielmann, niels, ceriel, fjseins, rkemp, versto}@cs.vu.nl
`http://www.cs.vu.nl/ibis`

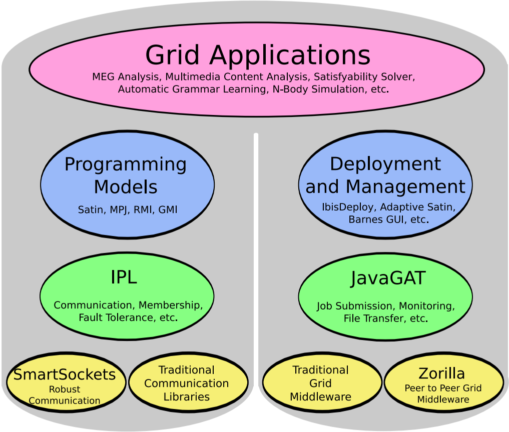
Although grids hold great promise for many scientific applications, writing efficient and portable grid applications is notoriously difficult. Grid programmers often have to use low-level programming interfaces that change frequently, and they have to deal with heterogeneity, connectivity problems, and fault tolerance. Also, managing a running application is complicated, because the execution environment changes dynamically, as resources come and go.

The Ibis project aims to drastically simplify the programming and deployment process of high-performance grid applications. The Ibis philosophy is that grid applications should be developed on a local workstation and simply be launched from there. This write-and-go philosophy requires minimalistic assumptions about the execution environment, and sends most of the environment's software (e.g., libraries) along with the application. To this end, Ibis exploits Java virtual machine technology, and uses middleware-independent Application Programming Interfaces that are automatically mapped onto the available middleware.

Ibis provides different programming abstractions, ranging from low-level message passing to high-level divide-and-conquer parallelism. All programming abstractions are implemented on the same Java library, called the Ibis Portability Layer (IPL). Ibis also includes a deployment system, based on the JavaGAT (which provides file I/O, job submission, job monitoring, etcetera, in a middleware-independent manner) and on the Zorilla peer-to-peer system. The Ibis system is designed to run in a "hostile" grid environment that is dynamic, heterogeneous, and suffers from connectivity problems. For example, Ibis solves connectivity problems automatically using a special sockets library.

User Base

Ibis software has been used to run many real-life applications like multimedia content analysis (University of Amsterdam), analysis of MEG images of human brains (VU Medical Center, Amsterdam), high-resolution video processing (the CineGrid project, based in San Diego), processing of spectroscopic data (AMOLF, Amsterdam), automatic grammar learning, SAT-solvers and many others. These applications are communication-intensive (and certainly not trivially parallel) and use multiple clusters at the same time, despite numerous connectivity problems. Ibis is also used by other European research groups to build high-level programming systems. The German D-Grid project uses Ibis to build a workflow engine for Astronomy; the University of Erlangen has built a grid file system on Ibis; the French ProActive system uses Ibis RMI to speed up communication; the University of Patras, Greece has implemented their Jylab system (a flexible scientific computing environment) on top of Ibis; HITACHI Europe (Sophia Antipolis, France) uses Ibis for peta-scale data management. Furthermore, Ibis has been used extensively on large Grid testbeds, such as the Dutch DAS-3, and the French Grid'5000 system.


Grid Applications
MEG Analysis, Multimedia Content Analysis, Satisfyability Solver, Automatic Grammar Learning, N-Body Simulation, etc.
Programming Models
Satin, MPJ, RMI, GMI
Deployment and Management
IbisDeploy, Adaptive Satin, Barnes GUI, etc.
IPL
Communication, Membership, Fault Tolerance, etc.
JavaGAT
Job Submission, Monitoring, File Transfer, etc.
SmartSockets
Robust Communication
Traditional Communication Libraries
Traditional Grid Middleware
Zorilla
Peer to Peer Grid Middleware

Current State

Ibis is an Open Source project, hosted at the VU University, and developed by over a dozen researchers and scientific programmers. A number of additional contributions were made by developers from all over Europe. These include work on interoperability with Globus 4 by members of the D-Grid project, and a programming model based on MPI by Siegen University, Germany.

The Ibis project started 6 years ago, and is a continuation of earlier Dutch BSIK and NWO projects, as well as the European Gridlab project. The continuation of the GAT functionality by the SAGA standard is supported by the European XtreemOS project and the CoreGRID NoE, as well as the OMII-UK initiative.

The Ibis project will be continued at least for the next several years. Ongoing developments include outreach to an even larger user base, in particular focusing on users who have limited or no expertise in large-scale distributed computing. This approach is exemplified by the Ibis tutorial, and our participation in SCALE 2008, both taking place at CCGrid 2008 as well.