Ibis: A Programming System for Real-World Distributed Computing

Henri Bal*

Vrije Universiteit
Amsterdam, The Netherlands
`bal@cs.vu.nl`

Abstract. The landscape of distributed computing systems has changed many times over the previous decades. Modern real-world distributed systems consist of clusters, grids, clouds, desktop grids, and mobile devices. Writing applications for such systems has become increasingly difficult. The aim of the Ibis project is to drastically simplify the programming of such applications. The Ibis philosophy is that real-world distributed applications should be developed and compiled on a local workstation, and simply be launched from there.

The Ibis project studies several fundamental problems of distributed computing hand-in-hand with major applications, and integrates the various solutions in one programming system. Applications that drive the Ibis project include scientific ones (in the context of the VL-e project), multimedia content analysis, distributed reasoning, and many others. The fundamental problems we study include performance, heterogeneity, malleability, fault-tolerance, and connectivity. Solutions to these fundamental problems are integrated in the Ibis programming system, which is written entirely in Java. Examples are: runtime systems for many programming models (divide-and-conquer, master-worker, etc.), a Java-centric communication library (IPL), a library that automatically solves connectivity problems (SmartSockets), a graphical deployment system, a library that hides the underlying middlewares (JavaGAT), and a simple peer-to-peer middleware (Zorilla). The resulting system has been used in several award-winning competitions, including SCALE 2008 (at CCgrid'08) and DACH 2008 (at Cluster/Grid'08).

The talk describes the rationale behind Ibis, the main applications (in particular multimedia content analysis), and its most important software components. Next, it discusses experiments where Ibis is used to run parallel applications efficiently on a large-scale distributed system consisting of several clusters, a desktop grid, and an Amazon cloud. Finally, the talk discusses recent work in using Ibis for distributed applications that run on mobile devices (smart phones) or that deploy distributed applications from a mobile device. An example is given where a distributed object-recognition application is launched from an Android phone, so the user can perform tasks that would be impossible on the limited resources of the phone itself.

For more information on Ibis, please see <http://www.cs.vu.nl/ibis/>.

* Invited Speaker.