Ibis: a Java-centric programming environment for computational grids

Henri E. Bal
Department of Computer Science
Vrije Universiteit, Amsterdam, The Netherlands
bal@cs.vu.nl
http://www.cs.vu.nl/ibis/

Ibis [vNMH+02] is a Java-centric Grid programming environment for distributed supercomputing applications. Java's high portability allows parallel applications to run on a heterogeneous grid without requiring porting or recompilation. A major problem in using Java for high-performance computing, however, is the inferior performance and limited expressiveness of Java's Remote Method Invocation (RMI). Earlier projects (e.g., Manta [MvNV+01]) solved the performance problem, but at the cost of using a runtime system written in native code, which gives up Java's high portability. The philosophy behind Ibis is to try to obtain good performance without using any native code, but allow native solutions as special-case optimizations. The pure-Java implementation of Ibis does several optimizations (e.g., compile-time serialization generation) using bytecode rewriting. The special-case implementations do more aggressive optimizations, even allowing zero-copy communication in certain cases.

The Ibis programming environment consists of a communication runtime system with a well-defined interface and a range of communication paradigms implemented on top of this interface, including RMI, object replication, group communication, and collective communication, all integrated cleanly into Java. Ibis has also been used to implement Satin [vNKB01], which is a Cilk-like wide-area divide-and-conquer system in Java.

Our current research on Ibis focuses on fault tolerance and on heterogeneous networks. We have designed a simple and efficient fault-tolerance mechanism for divide-and-conquer applications, based on redoing of work instead of checkpointing. The mechanism uses a global result table (similar to a global transposition table in search algorithms) to solve the well-known problem of orphan jobs. We have integrated this mechanism in Satin and showed that it works efficiently on two Grid test beds, the Dutch DAS-2 system and the (highly heterogeneous) European GridLab test bed. In addition, we have developed a new networking layer for Ibis called NetIbis that can run on a variety of (local and wide-area) networks by loading driver modules dynamically. NetIbis also provides an integrated solution to several connectivity, performance, and security problems from which many wide-area communication systems suffer.

References

[MvNV+01] Maassen, J., van Nieuwpoort, R., Veldema, R., Bal, H., Kielmann, T., Jacobs, C., und Hofman, R.: Efficient Java RMI for Parallel Programming. *ACM Trans. on Programming Languages and Systems*. 23(6):747–775. November 2001.

[vNKB01] van Nieuwpoort, R., Kielmann, T., und Bal, H.: Efficient Load Balancing for Wide-Area Divide-and-Conquer Applications. In: *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*. S. 34–43. Snowbird, Utah. June 2001.

[vNMH+02] van Nieuwpoort, R., Maassen, J., Hofman, R., Kielmann, T., und Bal, H.: Ibis: an Efficient Java-based Grid Programming Environment. In: *ACM JavaGrande ISCOPE 2002 Conference*. S. 18–27. Seattle, WA. November 2002.