

ARCHITECTURAL CONSIDERATIONS WITH DISTRIBUTED COMPUTING

Yibing Wang, Robert M. Hyatt, Barrett R. Bryant

Department of Computer and Information Sciences

The University of Alabama at Birmingham

1300 University Boulevard, Birmingham, AL 35294, USA

{wangy,hyatt,bryant}@cis.uab.edu

ABSTRACT

We understand distributed systems as a collection of distributed computation resources that work together as one harmonious system. It is the great achievement of computer networks, data communication and small but powerful computers that make it possible to use distributed commodity computers to facilitate distributed applications, to achieve reliable or high performance computing. The IT industry has been witnessing distributed computing progress from homogenous system to heterogeneous systems, with the scale evolving from LAN to WAN and then to Internet/Intranet. On the other hand, the increasing number and scale of IT applications have been posting new challenges to the academic field. Software approaches to distributed computing include client/server model, middle-ware components, agents, etc. System approaches include LAN, WAN, PVM (Geist et al, 1994), Cluster (Pfister, 1998), etc. While distributed computing at the system level has used open architectures for a long time, it remains immature in this respect on the software side. New distributed applications should follow open architectural standard to survive a longer time, and architectural considerations should be studied in the academic and engineering fields.

Instead of initiating a universal architecture for IT applications, too ambitious a task for a single group to achieve, we will first discuss some technologies that have been used in distributed computing based on our own experience. Through the advantages and disadvantages analyses of these technologies, we can have a better understanding toward architectural considerations in the networking environment. Then we will discuss distributed computing in the open networking environment (ONE), in which we are interested in studying the possible high performance, scalable and secure applications with inter-operable architecture.

Here we have three applications from previous works: a distributed MIS for post pivot (referred to as system1), a distributed component system for telecommunication management network (referred to as system2) and a dynamic stock information system over the Internet (referred to as system3). System1 has a client/server computation model with distributed databases. We had to build our distributed database system out of some commercial RDBMS products that support client/server computing but do not have distributed architecture themselves. That means the application software has to take care of many issues that come with distributed databases. Adopting distributed database architecture was for practical reasons. It turns out later that the software becomes very complex. Another lesson is that client/server is a simple solution for distributed computing over the network. However, it can be an obstacle to inter-operability and scalability because of explicit I/Os and ad hoc application protocols, if the system is directly built on low level networking facilities such as sockets.

Software components have been developed to address such kind of problems as well as for the re-use purpose. The communication middle-ware in system2 is an example of successful software components. Besides the functional independence requirement, architectural consideration is so important that it is the only guarantee that a component can be integrated into the target applications easily. What's more, there seems always a conflict between the ideas of software components and

performance. Software components need functional encapsulation and isolation. This will bring a lot of overhead in a component-based software system.

The conflict between distributed computing and performance also exists in other aspects. System3 was built with multi-thread architecture and memory based database. Performance was the key factor in its architectural design. However, the socket-based communication between the application server and clients has direct conflict with the parallel kernel. The socket connection becomes a bottleneck. Adding more computers in the system can not transfer the application server's tasks and distribute them among the new hosts, because the multi-thread kernel can not go across a single computer.

Understanding the fact that existing distributed and parallel computation models were developed in different problem domains helps in architectural design for large-scale software systems in the open networking environment. We are interested in achieving high performance as well as inter-operability and scalability in ONE. The research of distributed computing in ONE is well related to some other existing standards or ongoing projects, but there are differences in many ways. HTTP (RFC1945, 1996) and HTTP-NG provide the mechanism of information sharing over the Internet; while we are trying to design applications that can provide powerful, efficient and economic backend solutions. CORBA (OMG, 1999) provides a standard for component inter-operation and service invocation, but it is not a computation bound model for application servers. PVM is a software package for parallel computing in heterogeneous network environments. PVM depends on dedicated host computers to achieve high performance. Its task distribution strategy will not work well in an environment where non-PVM tasks exist and compete for CPU time with PVM tasks. Another drawback is its message passing technology. The packing and unpacking interfaces are not acceptable for general-purpose information transmission. MPI (MPIF, 1997) provides a standard message passing API. However, MPI is based on homogeneous platforms. Cluster technology also comes out of performance considerations. Clusters as a computation unit can be part of our open network environment, but the underlying network inside the Clusters may have to be private to guarantee performance. Thus, the interface between Clusters and ONE needs to be investigated.

With the open networking environment available and mission critical business application requiring, inter-operability, scalability and high performance software are needed. These requirements are self-contradicting from the point of view of present understanding of distributed/high performance computing and technologies available. But this is not necessarily true if we do take those challenges into consideration in the architectural design. We have observed some interesting facts among the applications in ONE. There is the possibility of parallelism under general distributed architecture (Wang, Hyatt & Bryant, 2000). We can use distributed and parallel computation techniques at different architectural levels. New computation schemes also apply.

Key words: Architecture, Distributed Computing, Open Networking Environment

BIBLIOGRAPHY:

1. Geist, A. et al, 1994, *PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Networked Parallel Computing*, MIT Press, Cambridge of Massachusetts, U.S.A.
2. MPIF, 1997, *MPI-2: Extensions to the Message-Passing Interface*, University of Tennessee, Knoxville of Tennessee, U.S.A.
3. OMG, 1999, *The Common Object Request Broker: Architecture and Specification*, Version 2.3.1, The Object Management Group.
4. Pfister, G.F., 1998, *In Search of Clusters*, 2nd ed. Prentice Hall PTR, New Jersey, U.S.A.
5. Powell et al, 1991, *SunOs Multi-thread Architecture*, USENIX - Winter'91, Dallas of TX, U.S.A.
6. RFC1945, 1996, *Hypertext Transfer Protocol -HTTP/1.0*, Network Working Group.
7. Wang, Y., Hyatt, R. M. & Bryant, B. R. 2000, *Software Parallel Architecture in Open Networking Environment*, (Abstract), to appear in "The Journal of the Alabama Academy of Science 71, 2" (April 2000), Alabama, U.S.A.
8. Zomaya, A. Y., 1996, *Parallel and Distributed Computing Handbook*, McGraw-Hill, Inc.