

An Update on CORBA Performance for HPEC Algorithms

Bill Beckwith

Objective Interface Systems, Inc.

Email: bill.beckwith@ois.com

CORBA technology today surrounds HPEC-oriented subsystems. In recent years CORBA is getting used inside those subsystems, but mostly to facilitate communication with nodes outside the subsystem. It is now possible to implement CORBA ORBs that have the performance characteristics required by HPEC applications. This talk will describe the effort to modify the OMG CORBA specification to accommodate HPEC requirements, one HPEC ORB implementation, and preliminary measured performance data.

CORBA provides a standards-based middleware architecture for building flexible distributed systems. The time-to-market and engineering life-cycle benefits of using CORBA in enterprise, server, and desktop systems are well documented. At the very least, embedding CORBA in very high-performance and parallel computing environments offers seamless connectivity to external environments such as Java virtual machines, web-integrated application agents, etc. Beyond the basic value, potential exists for building time-critical, data-intensive applications with the more flexible CORBA programming paradigm where communication is handled by a highly specialized Object Request Broker (ORB).

The determinate of this potential is the performance impact of using ORB technology on these time-critical, data-intensive applications. With the historical performance of desktop ORBs we often expect the CORBA GIOP protocol, and thus ORBs by nature, must add significant overhead to simple communications.

Thus, the wide-spread acceptance of CORBA use in very high-performance and parallel computing environments is predicated on the existence of specialized ORB technologies that can achieve nominal latency and optimal throughput consistent with or better than other parallel computing middleware technologies.

A useful timeliness measure of any ORB technology is the degree to which the use of the ORB is transparent with respect to the application performance. A parallel-computing ORB that provides a highly efficient use of the latest interconnect hardware makes the use of the ORB temporally transparent to the application.

The performance threshold that makes an ORB technology temporally transparent to applications depends on the nature of the application. An ORB that adds hundreds of microseconds to a small message transmission would not offer temporal transparency for most applications, but might offer temporal transparency for systems with less stringent requirements. If, however, an ORB technology can achieve performance by directly accessing hardware features of a high-speed interconnect that surpasses the performance of typical direct usage then the ORB become practical for all parallel-computing application uses.

There are two elements of the performance of a communication technology that are useful to this discussion: latency and throughput. Latency is typically measured in fractions of seconds and represents the time delay from when the sending begins to initiate a data transfer to the point at which the recipient starts to receive that data. Latency is easily benchmarked by measuring the end-to-end time to deliver small messages. Throughput is typically measured in bytes per second and represents a measure of the utilization of the underlying hardware's communication bandwidth.

A performant ORB technology would ideally:

(1) add little latency to low-latency, high-speed interconnect technologies for small messages, and

(2) add little or no overhead per byte transferred to high-speed interconnect technology for large data transmissions (i.e. minimal bandwidth reduction).

These two metrics are nicely represented on a X-Y graph where the X axis is data transmission size and the Y axis is total transmission time. A useful latency metric is the Y intercept of the line. A useful metric for throughput is the slope of the line.

The coordinated engineering of a highly efficient ORB implementation and high-speed interconnect hardware can offer application architects performance superior to the alternative of custom-designing the application. However, this superiority is only possible if the ORB technology is purpose-built and very closely integrated with the high-speed interconnect hardware.

Additionally, since such an application is using standard conformant APIs the application is portable to environments other than the performant high-speed interconnect and future generations of high-speed interconnects.

Latency and throughput provide only a piece of the performance puzzle. The correctness of many real-time applications depends on the predictability of the latency and throughput. A full discussion of optimality criteria is beyond the scope of this discussion but is important to application engineers building these systems and as a design constraint on the underlying ORB technology.