The Papyrus Integrated Data Server

T. Connors, W. Hasan, C. Kolovson, M.-A. Neimat, D. Schneider, K. Wilkinson
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304-1181
lastname@hplabs.hp.com

For the last two decades, the database community has been pushing a one engine solution to its customers. This position is reinforced by the acceptance of SQL as the standard DBMS interface language and of the relational server as the standard DBMS engine. The one engine is expected to serve all data-centered applications regardless of the data types managed, the way in which they are used, the performance requirements and the target hardware configuration. This position is as credible as having Detroit announce that one engine has been designed for all terrains and all vehicles, be they trucks, buses or race cars. In the meantime, many application developers have found that the one engine is no panacea, and have built their own specialized data managers. For example, CAD systems, with their need for complex object support, cooperative transactions and high performance, seldom use standard DBMSs for data management. They, as many other applications, have opted for specialization. Specialization provides enhanced specific functionality and improved performance, but it does so at the expense of multi-functionality.

Sophisticated applications often require the management of numerous and varied data types. For example, the need for multiple specialized data managers arises in multimedia systems that must manage text and voice. Similarly, Geographic Information Systems must manage map and attribute data. In such cases, the interoperability of the data managers must be handled in the application. This can be difficult if, for example, the various data managers use different transaction management conventions. Integration can also result in performance degradation because of the data passing and the coordination that must take place in the application. Thus, while the specialization of data managers is driven by functionality and improved performance, their subsequent integration can be difficult and can result in performance degradation.

We focus on the performance of integrated specialized data managers. In particular, we focus on the customization of parallel executions of data manager operators in a variety of computer configurations. This is done by specifying the glue that connects data manager operators in a way that is independent of the computer configuration; and then providing the ability to transparently target the execution to a variety of computer configurations ranging from centralized to distributed environments wherein parallelism and distributed processing are exploited. Parallelization of data manager operators built using Papyrus services is a challenging problem, but a more challenging problem is the parallelization of data manager operators that are built independently of Papyrus and are therefore black boxes to Papyrus.

Papyrus is a set of modules and services that enables the parallelization and integration of specialized data managers into one execution environment. Papyrus programs can be transparently targeted to different hardware configurations and can dynamically adjust at runtime to the number of available resources. A Papyrus System consists of a number of clients interfacing to a Papyrus Server. The Server consists of several integrated data managers executing on a multiprocessor system.

References

[1] T. Connors. *The Papyrus Object Library.* Technical Memo HPL-DTD-91-3, January 14, 1991.

[2] T. Connors and D. Schneider. *The Papyrus Execution Engine.* Working Paper.

[3] T. Connors, W. Hasan, C. Kolovson, M.-A. Neimat, D. Schneider, K. Wilkinson, *The Papyrus Integrated Data Server.* Technical Memo HPL-DTD-91-15, May 16, 1991.

[4] S. Ganguly, W. Hasan, R.Krishnamurthy. *Multiprocessor Query Optimization.* Working Paper.

[5] W. Hasan and S. Chaudhuri. *Pipelined Execution in an Extensible DBMS.* Technical Memo HPL-DTD-91-10, February 22, 1991.

[6] W. Hasan. *PIL.* Working Paper.

[7] C. Kolovson, C. Nguyen, M.-A. Neimat. *The Papyrus Record Manager.* Working Paper.

[8] M.-A. Neimat and K. Wilkinson. Extensible Transaction Management in Papyrus. *HICCS*, January 1990.

[9] K. Wilkinson and M.-A. Neimat. Maintaining Consistency of Client-Cached Data. *VLDB*, August 1990.