Data for the Masses

by Joan Peckham
University of Rhode Island

Database practitioners, programmers, and researchers have long dealt with issues arising from the complexity of an environment in which shared and persistent data is required. Many hard lessons have been learned and approaches developed. The new computing world order has many of the same issues and elements. However, there does not seem to be much transfer of knowledge from database theory and experience to these new systems and environments. Here we discuss the database practices that might be applicable to modern web-based and networked systems and how this information might be transferred to the general computing environment.

Fans of the television program, Star Trek already know that databases are the systems of the future. What further proof is needed than the mode by which the characters of the show interact with their computer, invoking the system’s name, “Database”. University students in their first database courses are introduced to a robust computing environment in which they must conceptually integrate all that they have learned about design, operating systems, programming languages, data structures, performance tuning, human interfaces, etc. All of these techniques become more complex, thus interesting, in the context of the database environment due to two simple assumptions relative to the general programming environment. They are: 1) the data persists beyond each individual computation, and 2) the data is concurrently accessible to many processes.

The transition from central batch processing systems to computers-on-every-desk has freed the individual operating system from some of the complexity of time-sharing models. However, with the connection of desk-top computers to local and wide area networks, new complexity has been created from the introduction of shared data and processing facilities. In this environment, the individual operating system becomes a slave of the networked/distributed system. This situation of shared and concurrently accessible data allows us to view the web as one very large distributed database.

Database experts have always explored ways to address the management of data and processes in a variety of environments. The database community is accustomed to tweaking assumptions and providing solutions to address these alterations. However, we have not been as good at sharing these results with other researchers and the general public. This gives rise to a repeated “invention of the wheel” to provide solutions and approaches for modern web-based systems.

The Problem

Nothing illustrates the problem better than a session on the web in search of a simple piece of data. Let us consider the following query:

“Find the primary web page for Person XYZ”

If we do not know XYZ’s institutional affiliation, then it is likely we will search for her using XYZ in our favorite search engine. If XYZ is active in her enterprise, then it is possible that over 1000 results (or hits) might be returned by a given search engine. This leaves the end user to do a linear search of the list of “hits” in hopes of finding the URL of the web site. Although search engines have improved greatly over time, this type of inefficiency still exists on the web. As a result, there are businesses that devote human time to search the web for requested information (HumanSearch 1998).

Arguments to explain the current state of affairs on the web include:

1) We are fortunate that database experts did not get involved in its conception as the rigidity of previous solutions might have biased the formulation of the web to some extent.

2) The best strategy for the development of the internet is to just throw out a loose prototype and permit users and researchers alike to propel its evolution to the best possible architecture. In this model, computers systems are living entities, capable of adapting and evolving with the ad hoc assistance of the masses of users and programmers.