SERVING UP REPORTS

By Mike Hill

There is a gaping hole in the client/server story, a void big enough to drain your Information Systems' budget and frustrate all your users' expectations. Call it "Reports".

Two years ago, a mining company hired an established consulting firm to upsize its financial system. Following a perfect six-month delivery schedule, report design started. Report programming completed 18 months and $500,000 later.

Last year, an apparel industry leader replaced all its business systems before defining the reporting needs. Now, month-end closing and payroll runs delay mission-critical reports for six days every month. One marketing report locks the user's work station for 4 hours before it completes.

A major environmental services corporation now debates moving a critical analysis system to its client/server network. Timely reporting is crucial to meeting environmental compliance regulations, and the current mainframe-based application meets this requirement adequately. Uncertainty over client/server report management governs the discussions.

Reports represent the basic products of all business systems. Despite the promise of "paperless" technologies, report generation cannot be reengineered out of the business process cycle. Failure to plan for managing the reporting process, preferably early in the client/server life cycle, results in either expensive retrofitting or lower performance and productivity. Instead of managing your business process, your company becomes managed by it.

Several large systems integration firms tout their client/server expertise, but cannot describe their report management strategies. Instead, they hawk the latest decision support and ad hoc query technologies. However, insufficient capabilities exist to process and display significant volumes of data.

Similarly, vendors of application development tools point to report writer programs in their product suites. However, these programs focus on client computing while neglecting the server side. Inefficiently served processes perform poorly in client/server environments.

What is needed is a strategic methodology for systematically handling all reports. Such an approach would take advantage of server based efficiencies to maximize system performance and user productivity throughout your organization.

FRESH APPROACH

Serving reports requested by client applications increases system performance as well as end user productivity. A *Report Server* system provides a simple approach for centrally managing high volume relational database reporting throughput for distributed computing systems. In addition, users directly monitor the progress and control the disposition of lengthy output products.

A Report Server services all reports for a company's entire database. The software supplies a straightforward mechanism that selects, starts, monitors, completes, reviews, and distributes reports. It consists of a *Report Server Engine*, a database interface, specific *Report Generator* programs, and *Report Request* and *Report Control* graphical user interface screens. The system runs continuously in the background on the database server's UNIX host platform, and is available immediately when the database system boots. It also directs reports and log files to common disk storage areas.

PROVEN SOLUTION

The Texas General Land Office and Texas Natural Resource Conservation Commission in Austin, Texas, recently deployed Report Servers on their large client/server installations. Successfully implemented using INGRES databases, these servers run on IBM/AIX, HP/UX, Sun Solaris, and Pyramid DC/OSx host platforms.

At the outset, programmers created reports using off-the-shelf tools not well suited to the client/server computing environment. There had been no prior analysis or planning for systematizing reports arising from the regulatory processes. Report generation programs lacked standardization, report screens locked until reports completed, poor performance plagued application servers, users' home directories held report files, disk space was at a premium, and users could not locate their reports when they did not print out as expected.

The systems realized overall performance gains after Report Server deployment, resulting from better resource utilization. Shifting report processing to batch mode on the central database server accomplished two things: application response time improved noticeably and network traffic decreased by half. Automated report file management resulted in more efficient disk storage utilization and relieved the system administration staff of trivial and time consuming maintenance tasks.

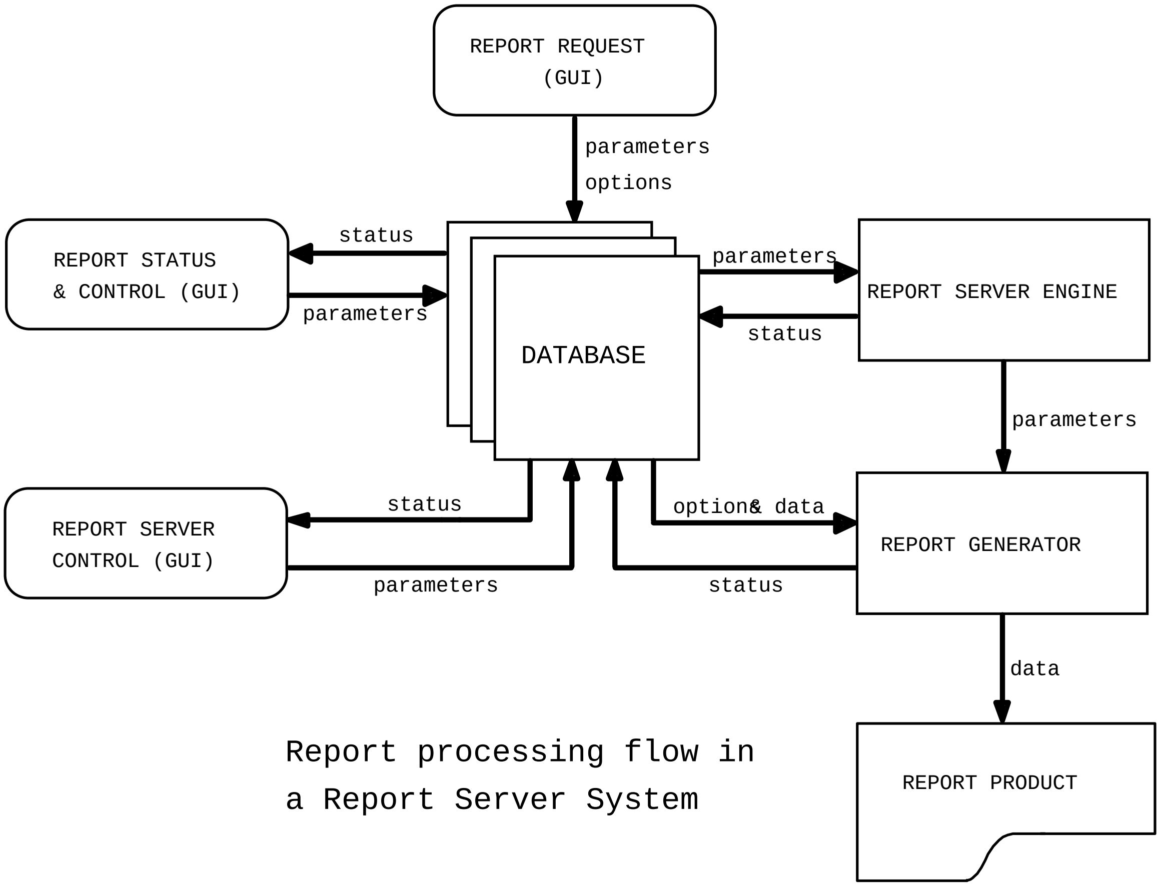
A capability allowing users to interactively preview reports contributed to network traffic reduction. Report preview also reduced loads on peripheral system resources, cutting wear and tear on printers and conserving ink and paper. Additionally, this feature eliminated users' waiting for hard copy to verify satisfactory report output.

However, the principal benefit came from users knowing when each report finished and where to find it; or whether it failed and why. Also, since processing shifted to the background, users could proceed with other work during report generation. These improvements dramatically reduced calls for assistance to help desk and programming support staff, and shortened response times to public requests for information.

HOW IT WORKS

The Report Server manages database report requests, runs report generation programs, tracks the progress of report generation, and distributes the output products. The basic report processing strategy consists of the following steps:

1. The user selects a report through a Report Request screen, which passes processing parameters and options to the database.
2. The database signals the Report Server Engine that a report has been requested.
3. The Report Server Engine reads the processing parameters, schedules the appropriate Report Generator for batch execution, and updates the report processing status.
4. The Report Generator reads the processing options, extracts and formats the data, and updates the report's status.
5. The Report Server Engine monitors and verifies Report Generator execution, and updates the report's status.



Report processing flow in a Report Server System

The overall report system revolves around interprocess communication through the database. Specific tables hold report processing parameters, processing options, processing performance histories, available reports, and available destinations.

A specific Report Generator produces each report. Running in the background, this program retrieves its report options, applies the options to its processing, and updates the report status. Processing consists of extracting the necessary data from the database, formatting the data for output, and sending the formatted data to a report file.

The Report Server Engine periodically checks that the Report Generator is still running, and assigns it a failed status when it does not complete normally. The engine also performs report system maintenance when it is not processing reports. It routinely handles report suspensions, cancellations, and deletions. Upon expiration of report retention limits, it collects processing performance statistics, removes files, and deletes processing parameters and options.

End users review the status and disposition of their reports through a *Report Status and Control* screen, which displays each report's name, file name, status, time started, time finished, and destination. Users may view, print, cancel, restart, and delete selected reports. Users also may select a system or local printer for hard copy distribution.

The database administrator monitors the active report processing load through a similar *Report Server Control* screen. At any time, the DBA can suspend report processing, or stop and restart the server, independently of other system and database services.

SIMPLE IMPLEMENTATION

Report Server implementation is easy. Generic program structure and standard SQL calls allow installation on a variety of platforms and databases with minimal change. Setting up common file storage is a trivial task. Customization adapts the Report Control screens and print queue interfaces for the system installation. A data interface facilitates ready integration with a print server.

Addition of a Report Server enhances existing application systems without replacing anything. Conversion and development of reports employ a simple programming interface that links Report Request screens and Report Generator programs to the Report Server system. End users continue using the same screens, performing the same procedures.

LOW COST, BIG GAINS

The only changes that end users will notice will be the speed at which their reports come back and the freedom to use their console time more productively. System administrators will note easy installation, network and disk utilization efficiencies, and less frequent system tuning. Database administrators will see increased query response times. Help desk and system analysts will enjoy having fewer problem calls. Managers will sigh with relief due to lower implementation costs and higher user satisfaction rates.

Reports continue to be the major products of computing. The advent of newer interactive technologies has not replaced the effectiveness of reports for presenting massive quantities of information for analysis, regulation, or decision-making. Reporting efficiency is key to having productive system users. Report Servers offer the necessary efficiencies for managing reports in large distributed systems with high utilization. Serving up reports to client users should be a part of the total client/server system solution.