

WHITE PAPER

WHITE PAPER



INFORMATICA

Analytic Applications: The Need for Scalability

Prepared for Informatica Corp. by
Tony Baer, Principal
Demand Strategies



Analytic Applications: The Need for Scalability

Introduction

Over the past decade, enterprises have invested tens of billions of dollars upgrading their core transaction systems to distributed, open system architectures. These migrations were driven by a mix of reasons that were competitive and defensive. Some enterprises were motivated by competitive desire to add new functionality, especially in supply chain management. Others were pushed into migration because of the need to achieve Y2K compliance or to find a less costly alternative to maintaining heavily customized, but poorly documented and rapidly aging systems.

As we reach the millennium, changes brought about through deregulation, mergers and acquisitions, and the spread of the Internet are driving many organizations to tap their IT systems for competitive advantage. This has spawned the need for a new generation of *analytic applications*. These applications bring to decision support what integrated enterprise systems brought to core transaction processes: the ability for the enterprise to act as more than the sum of its parts.

Drawing on vast stores of data, analytic applications help enterprises decipher how market demands or the identities of competitors are changing. Analytic applications can be used to correlate broad swings in product sales with customer satisfaction levels within individual demographic groups, or to discover new insights for improving the speed, efficiency, and quality of internal operations. While transaction applications allow enterprises to function, analytic applications direct them on how to navigate changing markets.

Analytic applications are decision support *solutions* to strategic business challenges. They pick up where data warehouses and data marts left off in that they are built to address specific business issues, rather than to create an infrastructure.

Analytic applications are emerging thanks to a new generation of capabili-

ties embedded in operating systems and databases, and delivered through third-party solution providers. Major software vendors, from Microsoft to SAP and hundreds of niche providers, are supplying a broadening array of offerings ranging from tools to packaged solutions for analytic applications. Increasingly, analytic applications are being adopted as *enterprise* solutions that leverage enterprise transaction systems.

As analytic applications are becoming ready for prime time, organizations must confront one hard fact of life: They must meet the same scalability criteria now expected of enterprise transaction systems.

Scalability must be planned for, even as early as pilot project phase. As analytic applications expand or change focus, the last thing that administrators need is to embark on costly reengineering efforts to cope with the consequences of growth on the underlying systems.

Business analysts and IT administrators alike must plan for the effects of growth, including growing utilization, subject focus, and analytical and reporting capabilities. It requires that attention be paid to the logical and physical aspects of the decision support system infrastructure. The consequences of ignoring scalability issues include deteriorating performance, poor quality data, and obsolete content.

This paper outlines the need for analytic application scalability, and discusses the technologies and disciplines that are necessary. It examines what scalability means for analytical system infrastructure, why it is important to system managers, and what are the technology and management ingredients necessary to make analytical systems scale.

What is Scalability for Analytical Systems?

The obvious answer is that scalability is all about maintaining consistent performance as systems grow in size. The obvious places to pay attention are the database, platform, and network because I/O and network congestion are the obvious causes of poor performance. For analytical systems, there are yet additional choke points: data integration and analytical processing—because the amount of data that are involved in both cases can be huge.

“When answering the need for decision support, it is often tempting to build analytic applications rapidly without paying attention to building sustainable infrastructure. This strategy begins costing hard dollars the moment the analytic application must be changed to add new functionality, add new data sources, or address new audiences. The “quick build” strategy quickly hits the wall.”

Although usually associated with physical performance, scalability is in reality a management discipline for coping with the requirements of growth and change. It is the ability to deploy a system without having to replace tools, technologies, or management processes as audience, database, or application scope grows. Performance is obviously a key part of the problem; effective management is the solution.

Lacking management discipline, the only alternative is to throw horsepower at the problem—such as increased processing power, more database tables or indexes, or additional business views. Such brute force responses are at best short-term solutions, that will resolve performance issues only until the next “crisis” emerges. They suboptimize the analytical system infrastructure for today’s problem, but do not anticipate what might happen if content changes, new competitors with new capabilities enter the market, various workgroups relocate to remote sites, business shifts to new product mixes or target markets, or sales territories are realigned. Short-term fixes cannot answer the question of how to cope with growing usage, application scope, and change without reinventing the wheel.

Managing scalability begins with specifying an architecture that can start small but grow over time. There are physical and logical aspects to architecture—both of which must be coordinated with one another.

Physical Architecture

The physical architecture involves the specification and sizing of servers, clients, databases, applications software, and in some cases, network capacity. In order for an application to scale, it must support the databases and platforms required for the group or enterprise. Ideally, deployment should follow an open systems model that supports common enterprise platforms, ranging from UNIX or NT servers, and Windows or web-based clients to allow users to take advantage of the broad array of third party tools and solutions available today.

To provide headroom for growth, the physical architecture must support a migration path to multi-processor, multi-threaded processing for all phases of the decision support application, from migration to everyday operation.

Logical Architecture

The physical architecture is developed in conjunction with the logical architecture—the stage where designers determine database and application deployment. The key questions to answer when developing a logical architecture include:

- Deployment mode — should a client/server architecture be used, using familiar Windows clients housing much of the application logic or localized business views? Or, should a web thin client architecture be used? Should the organization use a mix of client/server and web tools to different classes of users?
- Should the application or database be distributed according to usage by workgroup, geographic region, or both. If so, how should the application or database be partitioned?
- How should the target database model be structured?
- How should back end functions such as extraction, transformation, and loading be managed? Is throughput large enough to require batch processing? Are there portions of the application that might require real-time data migrations, or can the need be satisfied through interactive drill-throughs for specific detail data from source systems?
- How large and diverse is the user base? Should different classes of users be granted different degrees of access to the information, and different levels of query and reporting capabilities?

Meta data—the “data about data”—codifies the logical architecture, or the “rules” of the analytical system. It covers back and front-end processes alike, ranging from variables governing the extraction, transformation and loading of data, to data models, business view specifications, query and reporting environment parameters, access control, and job scheduling. Analytical systems live or die by meta data; lacking meta data, administrators and designers are left guessing on how to accommodate changing business needs.

Why Scalability is Important for Analytic Applications

Most analytic applications begin life as simple data mart pilots and grow in stages. The initial pilot is built using tools for creating infrastructure for back-end data management and front-end query and reporting. The investment is dominated by tools, rather than solutions. Then, other workgroups learn of the success of the pilot and start building standalone spinoffs.

As organizations realize the value delivered by such first generation marts, they examine the potential of decision support analysis capabilities for the rest of the enterprise. This is where the scalability issue emerges.

While scalability has a common definition involving the ability to grow the size or throughput of a system without degrading performance, in operation, the stresses on an analytical system are far different from those of transaction systems. For instance, unlike transaction systems, the “size” of analytical systems can be difficult to predict. While transaction system sizes can be estimated based on factors such as sales, production volumes, or revenues, analytical system “sizes” are related directly to utilization rates and scope of data, subjects, business views, and application functionality. And those parameters can vary sharply.

For instance, in some organizations, usage often peaks with factors such as emergence of an urgent competitive challenge or the rate at which the company changes its product mix or sales territories. Another factor that complicates the challenge of estimating system size is that the raw processing associated with analytic applications ranges sharply in complexity and data throughput. Depending on the analysis, huge amounts of data may be required, which might be migrated from one or more source systems, or derived from multiple sources including an OLAP system, an operational data store, or via drill-through to legacy systems.

Consequently, data administrators need to be able to move the necessary data when it is needed—regardless of volume. Doing so can become a balancing act, because the decision support needs of different classes of users can vary. Consider the following slice of a typical enterprise:

- Sales analyzes which reps have done best with which products in which regions;
- Marketing examines broad demographic trends requiring multiple views of product families or types by target market to learn which products have sold to which customers, which customers are the most profitable;
- Operations requires trending data on manufacturing and distribution activities;
- Corporate finance requires consolidated views of expenditures by business organization.

The rate of refresh and usage of analytical systems for different parts of the organization may vary. For instance, while finance and sales may work on monthly and quarterly periods, operations may require weekly or in some cases daily snapshots, while marketing is likely to demand longer-term views.

The result is that the analytical system infrastructure must handle uneven spikes of activity, both in data movement and user reporting, that may peak at the end of a month or quarter—or at random, whenever market upheaval suddenly occurs. Given that batch windows in many organizations are getting shorter and shorter, data migration processes must be conducted as efficiently and rapidly as possible.

Additionally, while decision support needs across the enterprise may be diverse, in many cases, each end user constituency may use overlapping slices of data. For instance, sales and marketing may each rely on the same product data, but perform different forms of analysis or require different business views or degrees of access. Therefore, the logical architecture—the data models and the migration parameters—must be well coordinated to prevent duplication of effort which, for large analytical processing environments, could be costly. It requires close coordination of back-end data integration processes with front-end user tools.

Technology Requirements

When answering the need for decision support, it is often tempting to build analytic applications rapidly without paying attention to building sustainable infrastructure. This strategy begins costing hard dollars the moment the analytic application must be changed

to add new functionality, add new data sources, or address new audiences. The “quick build” strategy quickly hits the wall.

Therefore, building sustainable, scalable analytic applications starts with flexible deployment platforms – the core of which is an enterprise data integration hub that is responsible for extracting data from operational sources, enriching it for decision support, cataloging it for use and reuse, and delivering it to powerful business intelligence and analytic applications.

This deployment infrastructure is used to build enterprise analytic applications, which for some organizations might start with small, pilot data marts. As the marts grow successful, they often morph into enterprise data warehouses, or into a series of workgroup data marts—or a network of marts may be organized around a hub data warehouse.

Regardless of how it is designed, the data integration hub must address four key requirements that, taken in concert, manage all the key decision support processes. They can range from data migration to business view creation, and the management of query environments, servers, and networks. These four requirements are:

- Consolidated global configuration
- Operation management
- Transformation scalability
- Session performance

Consolidated global configuration

The enterprise hub must provide a consolidated view of data migration with a global configuration engine that governs all the necessary parameters of extracting, transforming, and loading data from source to target. The global view must encompass all sources and targets used with the analytical system. This is necessary, regardless of whether they are feeding an operational data store, enterprise data warehouse, or multiple data marts.

If the analytic application involves multiple marts deployed throughout the enterprise, the analytical system data migration software must provide a consolidated view for coordinating the loading of end user or workgroup marts. Lacking such capabilities, IT administrators will find themselves constantly repeating their

efforts, migrating and re-migrating overlapping slices of data.

The view of migration is driven by *meta data*, which is stored in a repository. This repository, in effect, provides the road map for analytic applications.

Depending on the organization’s preference, the repository may be centralized or distributed. The data migration engine must support the ability to operate in either mode: either a global or a hierarchical, federated system of repositories that can be managed from a single console, using a top-down or bottom-up view. If viewing from top down, the administrator must be able to click on a piece of global meta data, and have the ability to drill down to detailed data residing in the satellite repository; conversely, when viewing from the bottom up, satellite repositories should have pointers to the master.

When managing the system, any change made by the administrator should automatically percolate down. For instance, if the enterprise restructures itself from regional to product-based organizations, the administrator should only have to make the changes to data transformations once.

Using a scalable meta data repository, the administrator should also be able to manage end user installation and configurations from a single console. Single console views simplify the configuration of user access rights and security, which may vary depending on factors such as data content, query complexity, reporting length, time of day, and user role. For instance:

- Developers may enjoy full read/write access rights to meta data configurations for a single mart or group of marts, but read-only access to others.
- Product line managers may enjoy full access rights, and broader query and reporting privileges based on their role as manager for their own product family. For instance, they may be entitled to run more detailed reports than sales staff. However, for other product families, their access rights may be equivalent to that of sales staff.
- Individual sales representatives enjoy full access rights to data from his or her region, but only summary-level access to those of other regions.

The only practical way to manage large, diverse user populations is from the server. Otherwise, administrators are left dealing with multiple, standalone systems, which may result in the entry of inaccurate or inconsistent user access privileges.

Operation management

As with configuration, single console-based control is essential to prevent anarchy. This includes a global log of all analytical system processes, from the back end extraction, transformation, and loading operations to front end activities including interactive query and analysis for power users, along with the preparation of staged reports intended for casual users.

Depending on the organization, administration of analytical system operations may be the domain of data warehouse specialists, UNIX or database administrators, and/or IT systems/network administrators. Analytical system tools must provide the options for self-contained management, or integration with management processes with third-party management frameworks or point tools from Tivoli, HP, CA, BMC, Compuware and others.

There are many components that must be managed for optimizing performance of the analytical system, extending from the database to servers, storage subsystems, and networks. That includes balancing CPU utilization, memory caching, and disk swapping (paging data to off-line storage). Within the database, factors such as tablespace allocations, SQL activity, and I/O "hot spots" must be monitored and managed to prevent bottlenecks, or at worst, crashes. Additionally, coordination with network management systems is necessary, because processes varying from large, bulk migrations or peak-level, interactive processes could be hampered by insufficient bandwidth.

For the analytical system administrator, a single console view of all servers is necessary for coordinating the monitoring of operations, display of critical alarms (when allowable thresholds are exceeded), user access control and security management, and production scheduling.

Transformation scalability

There are several different data transformation approaches that are currently available off the shelf. Most

tools are based on specific programming languages; because much of the data is stored on legacy systems, COBOL-based transformation tools have been popular. The drawback, however, to language-based tools is that they provide hard-coded approaches to transformation that are time-consuming to maintain or modify. Given the reality that change is an ongoing process in managing analytic applications, this issue has become a serious drawback for language-based tools.

A newer generation of visual object-oriented data transformation "engines" have been proven as the fastest, most maintainable. New-generation, visual transformation engines provide a rich library of data-driven functions that can be activated or modified with the click or dragging and dropping of a mouse. Such approaches simplify frequently-used processes such as case or file conversion. You simply drag and drop the appropriate data elements onto the operation, rather than writing procedural code that must be debugged.

Additionally, the object-oriented nature of the new-generation transformation engines allows administrators to take advantage of capabilities such as inheritance. A classic example occurs when product families are changed; with an object-oriented engine, they can become "properties" that are automatically passed on to individual products.

The ease of use of engine-based transformations is a scalability issue because, in larger analytical systems covering multiple subject areas and end user constituencies, there are simply more transformation operations to be developed. Given a choice of writing several hundred transformation programs vs. dragging and dropping, which mode would you pick?

Session performance

As enterprises discover themselves competing in a 24-hour global marketplace, the available batch window for moving data is growing extremely tight. There are several key features found in leading data integration products that best utilize the limited batch windows:

- **Multi-Threading:** Enterprise data integration products should support UNIX multi-threading capabilities, and efficiently utilize multi-processor platforms. In this manner, a single time interval

can be used to support multiple concurrent processes. In effect, several sessions can be ganged at once.

- **Pipelining:** The migration process involves a series of individual read, transformation, and write steps. If processed sequentially, large data loads can get bottlenecked, because each of the steps may take varying amounts of time to complete.

Delivering Analytical System Scalability: Informatica PowerCenter

Scalability has become a major focus of data integration and transformation vendors. Informatica offers an approach that allows organizations to grow their analytic applications in several ways. They can be built in a phased approach, where multiple data marts are “federated” in a peer-to-peer arrangement to build an enterprise analytical system; alternatively, they can be grouped as satellites, providing local analytical capabilities feeding off a central source.

Demand Strategies believes that a phased approach is critical, because unlike transaction systems, most analytical systems begin life as pilots serving narrow constituencies. Another critical distinction is that analytic applications are often best built incrementally, where new functionality is added as end user constituencies grow more familiar with decision support capabilities.

To deliver scalable performance, Informatica PowerCenter provides:

- Support for leading UNIX and NT server platforms;
- A powerful, object-oriented, visual data transformation engine;
- Multi-threading support to take advantage of parallel processing features in operating systems and multi-processor platforms;
- A scalable repository system that supports global and federated repositories;
- An integrated control center for managing role-based security, end user access control, and resource/load balancing of key infrastructure features, including CPU, channel, memory, and disk subsystem utilization;
- An integrated performance monitoring and event logging console which allows analytical system managers to check performance and make configuration changes on the fly;
- Support of pipelining/streaming to optimize use of processor power during high-volume read/transform/write processes; and
- Native database drivers which take advantage of all the key performance features offered by major relational databases.

Pipelining (or “streaming”) is a special technique that keeps processors continually busy, by immediately moving processed data into cache if the next process is not yet ready. In effect, pipelining is a delicate balancing act that optimizes CPU processing with memory read/writes, internal (channel or bus) and external (network) communications.

- **Native Database Drivers:** While many tools support ODBC because it provides a common pathway to all major SQL relational databases, native database drivers offer superior performance. For instance, ODBC performance deteriorates when handling certain forms of DATE/TIME data, or for numbers exceeding 15 digits. Native drivers also allow you to perform high-speed bulk loads—an essential feature when millions of rows of data are involved.

In an ideal world, the central data integration product in an enterprise analytical system should be able to take advantage of all of these capabilities. In the real world, however, there are always limitations on processors, channels, memory, disk, etc. For instance, it is advisable to load bulk load the database directly, without going through a memory buffering stage that involves an extra I/O step. However, in some cases, hardware, processor, or connectivity limitations may preclude this.

Using an integrated control console, the data integration hub must provide a seamless means for balancing the utilization of different infrastructure elements, while determining which processes should be concurrent or sequential, and when the processes should be run.

Additionally, because of the requirements of data conversions, the hub should have a high-speed staging cache, which can be used for buffering data (if necessary) or providing a rapid means for validating conversions before the loading process begins.

The icing on the cake is the need to monitor each of the processes that migrates data from source to target. With a multi-step process, huge throughput, and the involvement of numerous elements, analytical system administrators need to understand where the bottlenecks and errors are occurring, or which processes can be more optimally tuned.

Ideally, the data integration hub should provide a means for making configuration changes on the fly—a critical function, given the changing nature of analytic applications, and the data that populate them.

Scalability Matters

Analytic applications are becoming the next generation of enterprise applications. With enterprise transaction systems providing a vast, integrated store of data, organizations can leverage their ERP investments with new solutions that help them navigate changing market terrains.

Because they are being deployed at the enterprise level, the systems infrastructure underlying analytic applications must provide scalability equal to or exceed that of the transactions systems which they are supplementing.

Scalability matters because analytic application users require performance, and in many cases, large amounts of aggregated data. With end user constituencies numerous, diverse, and often distributed across multiple geographic locations, performance is especially critical. The challenge is heightened by the high-volume data integration and migration used for populating analytical databases—and the fact that the available batch window for migrations may be extremely tight.



Analytic Applications:
The Need for Scalability
prepared for Informatica Corporation
by Demand Strategies
40 Putnam Road
Bedford, Mass. 01730
Tel: (781) 271-0010
Fax: (781) 271-1348
E-mail: tbaer@demand-strat.com

INFORMATICA®

For more information contact:

Informatica Corporation
3350 W. Bayshore Road
Palo Alto, CA 94303
Telephone: 650.687.6200 or 800.653.3871
Fax: 650.687.0040

Informatica's World Wide Web address is
(URL) <http://www.informatica.com/>.

Copyright © 1999 Informatica Corporation. All rights reserved. Printed in the USA.

Informatica, PowerCenter, PowerMart, MX and MX2 are trademarks of Informatica Corporation. All other company and product names may be trade names or trademarks of their respective owners.