AlphaBlox™

Making More from Less

Scaling the Analysis Server

SUMMARY

The analytic infrastructure—the broad, deep platform upon which decision-makers rely to identify, understand, and pursue their most profitable opportunities at the front lines—revolves around the analysis server. This server provides intelligence to intranet and extranet users through analytic applications, which combine information from multiple points across the organization's business chain, run it through analytics and customized business rules, and deliver it over a browser or other network device.

The rise of analytic applications brings with it a new problem: popularity. The viral dynamic of more users wanting more access to more applications on more data from more places in more time zones leads to increased burden on the analysis server. The organization weighs the wants of users and the realities of resources and explores the solution of clustered analysis servers to ensure the scalability needed to support the analytic infrastructure.

Sooner or later, all managers tasked with building or using the analytic infrastructure confront the business and technical issues surrounding scalability. What drives scalability from the business perspective? From the technical perspective? What is the most effective way to scale up the delivery of analytic applications? How does scalability affect the analytic infrastructure?

This paper will answer these questions by exploring the problem of scalability, outlining users' wants and IT's resources, focusing on the general solution of scaling with clustered analysis servers, and explaining its specific implementation using Alphablox.

THE ANALYSIS SERVER AND THE NEED FOR SCALABILITY

In the age of distributed, Web-based applications, users are no longer content to sift through large volumes of data to identify profitable business opportunities. With too much of the burden for mining and massaging the data placed on the user himself, printed reports (neither broad nor deep) and client-server business intelligence tools (broad, but not deep) have given way to the *analytic infrastructure*. Upon this infrastructure the organization assembles *analytic applications* which reveal the organization's most profitable transactions and which allow users to interact both broadly and deeply with a variety of data sources.

A typical analytic application runs on an analysis server, with which users communicate through a simple Web browser. For instance, an account manager might log on to an analytic application over the Web to review his sales by product. Delivering information from widely diverse data sources in the company, and applying the organization's customized analytics to that information, the application can help the account manager to drive his sales most profitably by answering questions such as:

- Which products generates the most profit?
- Where can I sell more units of that product?
- How can I make each unit cost less and generate more money?

The analytic applications serve the account manager and the other members of his sales team. Before long, his co-workers in Production see him using the applications, and are impressed by how they offer information on costs. They, and other users on the intranet, recognize the value of analytic applications, and the organization accommodates them, increasing the workload on the analysis server.

Suppliers, partners and customers—extranet audiences—also see the value of these applications. They want to use them on a business-to-business level, to schedule their own production, delivery, purchasing and payment. This requires more applications and modifications to existing applications to prevent access to proprietary information. And, with requests and pages flowing into and out of the company, the matter of network security becomes more acute.

The analysis server designed for 25 users must now meet the needs of thousands of simultaneous users and line-of-business (LOB) managers inside and outside the company. Both the hardware and the software components of the server are running at their upper limits, the applications are running more slowly, intranet and extranet users alike are less satisfied, and the analysis function has lost some of its former value because of the performance degradation. More resources are required for the analysis server to run properly; in short, a problem of scalability, with both business and technical dimensions.

SCALABILITY—THE BUSINESS PERSPECTIVE

"When it comes to business intelligence, scalability has many meanings because it is needed on many fronts in e-Business. Growing user communities, increasing numbers of reports, and burgeoning data volumes are driving a need for scalability in BI in general." (Hurwitz Group, Inc., Jan 2001)

The business perspective of scalability is as broad and expansive as the collective imagination of the user community, and its hallmark is the comment, "Just think what we could do if…" We explore five business drivers towards scalability: more users, more data services, more formats, more evolution, and more of the decision pushed to the server.

The Viral Dynamic: More Users in More Places at More Times

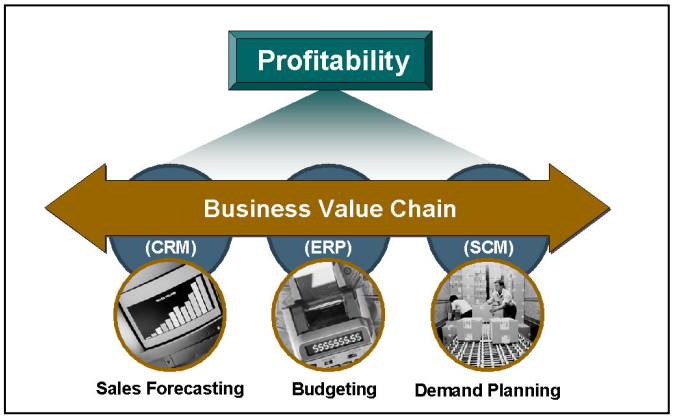
As we've seen, the value of the analytic application rises with the number of users and the amount of collaboration the application enables. If the Sales team has the information with which to make better decisions, it's only a matter of time before Marketing, Production and Operations begin to ask for similar applications. It's also only a matter of time before customers, suppliers and partners outside the organization see the mutual advantage in wider access to the application.

The application is of even greater value when it is geography- and network-independent. Users want access to the application—or at least to the intelligence it provides—in more places farther from the intranet: from home, over dial-up, offline.

With geography-independence comes time-zone-independence: The sun never sets on your analytic application because when Europe is sleeping, Asia is waking up. The scalability solution needs to provide for constant access to the analytic infrastructure, always providing the opportunity to identify the next profitable transaction, wherever and whenever it may arise.

Answers Drawn from More (and More-Varied) Data Services

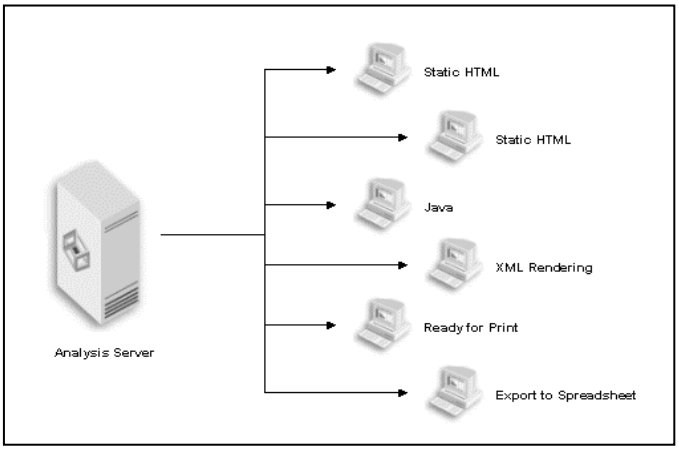
Once an analytic application answers sales forecast questions based on CRM transactions, for example, it begs budgeting questions whose answers lie in ERP transactions. Applications answering those questions then beg production questions based on SCM transactions and events.



Whereas data may formerly have come from a few similar sources through ODBC (Open Database Connectivity) or a relational database, a hypothetical application can now include MDDB (multidimensional database) or OLAP financial data from Essbase; supply chain data from Microsoft OLAP Services; ROLAP (relational online application processing) data for Bookings, Billings and Backlog; and RDB (relational database) demographic attribute data. The analysis server is central to accessing these different data types, running the data against analytics, and delivering pages based on these diverse sources.

Answers Rendered in More (and More-Usable) Formats

With geographic independence comes the variable of throughput. In its first release, an analytic application may assume that all users have broadband access to the system and need to analyze intensively; in later versions, the application accommodates many different levels of throughput and presentation.



The most versatile analysis server can present a single application in multiple formats for user-types who:

- are on the LAN, performing advanced interactive analysis with broad functionality atop Java applets
- are on a less robust network, performing some interactive analysis with interactive HTML
- connect via 28.8K dial-up and work in static HTML views of data
- want only to print a view of the data (e.g., as a portable document format, or PDF, file)
- work offline, with the data exported to spreadsheet format for later analysis
- receive alerts on client devices such as mobile phones and PDAs via XML

As users want related information rendered in multiple ways—for example, a sales application which is updated near month-end, plus a static PDF of eastern region reports from the application, plus a paging request if a supplier's price drops below a certain threshold—the analytics used to perform these calculations and renderings place new demands on the analysis server.

ANALYTIC INFRASTRUCTURE AND SERVER DISPLACING BI TOOLS

Evolving from BI tools to analytic infrastructure means evolving beyond many long-held techniques for delivering information to users. Query & reporting tools, spreadsheet models, graphic front-ends for OLAP, and even "integrated" suites of BI tools generally offer analytical value, but at the cost of a high-maintenance client and inadequate flexibility. Only an open, extensible, Web-based, analytic infrastructure can accommodate development needs, the inevitable growth of the user base and the increasing number of analytics in applications.

At the heart of the analytic infrastructure is the analysis server, and the enterprise that comes to rely on it will invest in its scalability.

MORE OF THE DECISION PUSHED TO THE ANALYSIS SERVER

The account manager whose specialty is selling, not hunting for data; the production manager who excels at just-in-time, not at pivot tables; the lab director who wants to manage projects, not query tools, all benefit from the intelligence delivered by analytic applications. In the next stage of eBusiness analysis, they will want their applications to find the most profitable transactions for them, and not to bother them at all with any unprofitable alternatives. And, once analytics have been tuned to their users' acceptance thresholds, some decisions will disappear altogether: the most profitable transactions will be executed automatically, with simple notification sent to the users.

With the server bearing an ever greater share of the decision-making process, client processing gets correspondingly lighter and lighter. With a wider range of solutions and formats (ready-for-print, export to spreadsheet) supporting users on low-speed connections, application traffic on the client decreases, and the workload on the analysis server increases, further driving the need for scalability.

SCALABILITY—THE TECHNICAL PERSPECTIVE

The unlimited wants of the business perspective meet with the resource limitations of the technical perspective. The key phrase is "Yes, but…" Technical impediments to scaling analytic applications are not always financial, either; in some cases, the software or hardware solutions do not exist, or it is not possible to implement them without unacceptable levels of disruption.

We explore five compelling technical issues with scalability: transparency, upgrading hardware, replacing hardware, flexibility even at entry-level, and security.

Increase in Capacity Must be Transparent

The first priority from a technical perspective is not to inconvenience users from system improvements from scaling. As the analysis server moves to center stage in the decide-and-act process, that server scales best which requires the fewest—or, preferably, no—changes in how the organization uses it.

The process of scaling up or down should not affect how users connect to and run their analytic applications on the server, how developers build and implement these applications, or how IT administers the server. The server will process page requests and perform its analysis more quickly, and users will not need to change the way they work.

Additional Processors on Existing Server

Upgrading existing hardware is a common response to the need to scale. More processors, more powerful processors, more memory, and greater network input/output are all options. For a departmental analysis server with growing needs, this may be a valid and cost-effective option.

Where larger numbers of intranet and extranet users are being served, however, the long-term demands on the analysis server may require more than additional processors or memory—these demands may require a server farm. If the organization has a server farm already in place, it may wish to dedicate some of the farm's capacity to analytic applications. This has the added benefit of avoiding a single point of failure for the analysis servers.

Maximum Flexibility even with Entry-Level Hardware

Scalability is also a concern for organizations on the verge of deploying their first analysis server. Still unsure where the effort will take them and at what speed, they frequently ask, "How powerful a machine do we need?" (i.e., "How small a server can we get away with?"). They want to begin with an investment commensurate to their current needs and add to it as those needs grow.

A proper solution for scalability will allow such a team to begin with the analysis server running on entry-level hardware and add capacity as workload and number of users dictate.

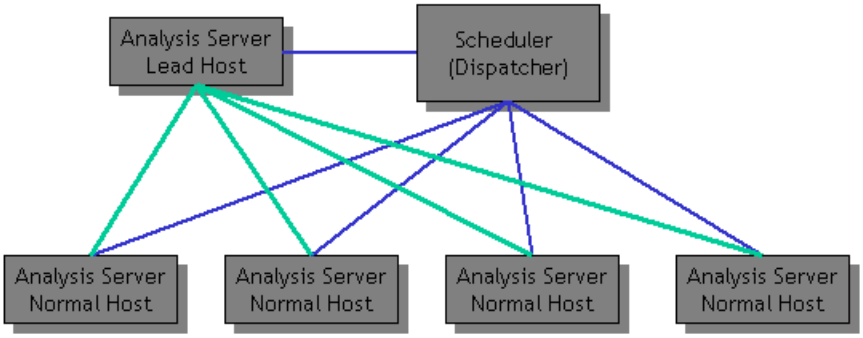
No Compromise to Security

Whatever solution may prevail, it must be compatible with existing levels of security and not expose transactions. Firewalls, access control lists, user and group privileges must all protect the scaled analysis server as well as they do the existing one.

Solution: Scaling via Clustered Analysis Servers of Varying Cost and Power

The technique of clustering analysis servers provides the advantages of scalability and reconciles the aforementioned business and technical concerns. It allows more users to run more analyses on more data as the analytic infrastructure expands. It provides:

- minimal disruption to users, developers and administrators,
- minimal change in how users connect,
- maximum leverage from entry-level hardware,
- and a gradual investment in the analytic infrastructure.



The solution consists of a cluster of Unix- or Windows-based servers of varying cost and power, connected over fiber or another high-speed network, each running its own instance of the analysis server. A *dispatcher* assigns sessions among servers (usually based on respective CPU load). One of the servers plays the role of *lead host*, tasked with replicating changes among the other servers (*normal hosts*). Although the dispatcher is the only server visible to users, sessions between a user and a server in the cluster continue normally. Thus, the clustered model shares workload among the servers, does not affect the users' experience, and allows for scaling the analysis server up or down as needed. This solution is also known as "load balancing", because it tries to equalize the incoming workload among multiple servers.

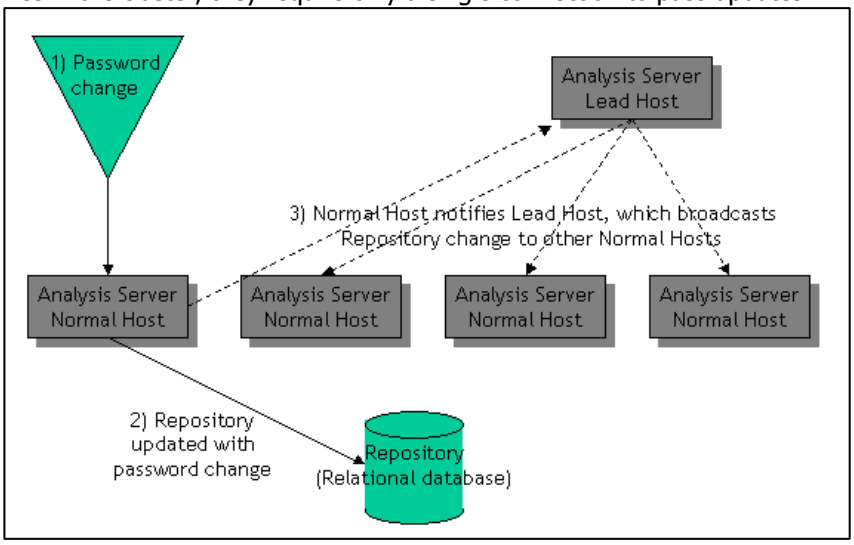
TECHNICAL REQUIREMENTS—ANALYSIS SERVERS

Aside from the analytic application queries and the result sets passing between data source and user, session-related data also move across the analysis server and reside in the *repository*. The repository is a database (a relational database, ideally) containing operational data about users, groups, objects, resources, and status of analytic applications.

With a single analysis server, changes to the repository affect only one machine, and no other machine requires the update, but with multiple servers in a cluster, it is critical that any changes (e.g., changes to user and group data, session status, application state, access control) move quickly to all machines.

ONE ANALYSIS SERVER MUST LEAD

If there are only two machines in the cluster, they require only a single connection to pass updates back and forth; with three machines, three connections; with four machines, six connections …and the number of required connections continues to grow to roughly n^2 for n servers. To avoid the proliferation of necessary connections among individual servers, one server can play the role of the "lead host" to the other "normal hosts," wherein a cluster of n servers



requires only *n* connections.

Each Server may Update the Repository

Since each server in the cluster is running an instance of the analysis server and of at least one application, each server needs to be able to communicate repository updates to the other servers. If, for example, a user changes her password during a session running on one server, the repository must be updated immediately and directly so that the new password is available to all servers in the cluster.

Each Analysis Server Communicates with Lead Host

As a server makes a change to the repository, it notifies the lead host of the change. The lead host then broadcasts the change to the other servers in the cluster. A cluster manager runs on each analysis server to handle these notifications, and also to broadcast its own server's current state (up, down, busy) to the lead host.

Repository Must be Relational

With the repository now handling updates from *n* analysis servers instead of from just a single one, the robustness of the repository becomes an issue. A relational database (RDB) repository—as opposed to a shared file system repository—allows locking and protection of repository objects, which prevents damage to the repository when multiple servers attempt to access the same object. Also, administrators can access, maintain and back up the repository using the more powerful tools common to relational databases.

Technical Requirements—Dispatcher

In the same way that the lead host handles updates to repository data among the analysis servers, the dispatcher maintains user sessions with analytic applications by routing them as IP (Internet Protocol) traffic between the clustered servers and the clients. For efficiency, the *only* way for the user to communicate with the analysis server is through the dispatcher.

Dispatcher

The dispatcher runs on one of the servers (or its own dedicated server) and routes incoming page requests to the least loaded server in the cluster. The dispatcher also routes subsequent requests to that same server, an important requirement for session continuity. As more analysis servers join the cluster, the dispatcher can begin routing incoming requests to and maintaining sessions with them as well.

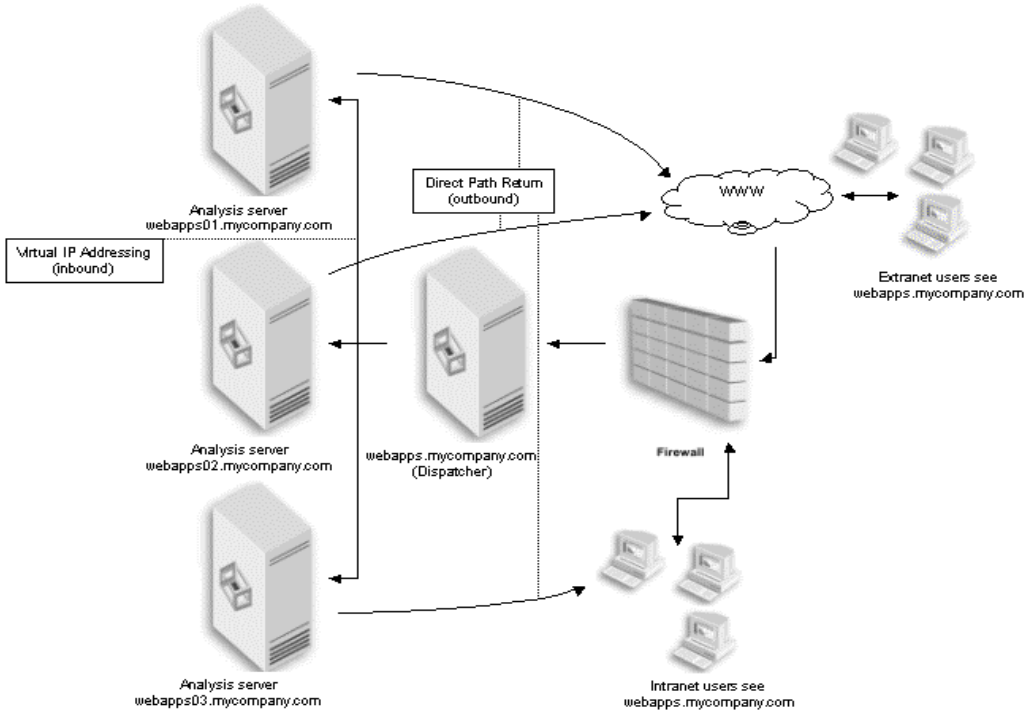
For additional reliability, administrators may configure one of the other servers as a backup dispatcher.

Virtual IP Addressing

In a 100% Web-based analytic infrastructure, users communicate with applications in the same way in which they surf the Web: by entering an address (e.g., webapps.mycompany.com) to their browser. The address points to a *single* server, and to the opening page of the analytic application. But in the clustered solution the user could be running the same application on any one of *multiple* analysis servers behind the dispatcher, each with its own private address (e.g., webapps01.mycompany.com, webapps02.mycompany.com…). How does the solution stay flexible and scalable while still allowing users to simply connect to webapps.mycompany.com, as they always have?

In order for the clustered analysis server to work efficiently and with minimal changes to the way in which users connect, the address webapps.mycompany.com points to the dispatcher *only*, so all users continue to connect to their analytic applications using the same address. The dispatcher then invisibly routes the requests to the clustered servers using technology called "virtual IP addressing" as part of

its load balancing. All of the thousands of connected users appear to be running applications on webapps.mycompany.com, but their sessions are in fact running on analysis servers behind webapps.mycompany.com.



Also, this offers a measure of security for the analysis servers in the cluster, because unauthorized external users cannot address them directly, and authorized users address them exclusively through the dispatcher.

DIRECT PATH RETURN

While all incoming traffic goes first through the dispatcher then to the analysis servers in the cluster, outgoing traffic from the analysis servers goes directly back to users, a technique known as "direct path return". This avoids the potential bottleneck of sessions moving both in and out via the dispatcher, while keeping the physical analysis servers invisible to inbound traffic.

PERSISTENT SESSIONS

With virtual IP addressing, the dispatcher routes traffic among various analysis servers, but if a session is already in progress between a user and one of the servers, it is critical for the dispatcher to keep the traffic moving *intelligently*, so that the session persists and is not interrupted. This persistence is essential for functionality as transparently interactive as being able to move backward and forward in the application through the browser—relatively simple in a client-server context, but more challenging in a 100% Web-based environment.

For the user's work to continue normally, the analysis server creates a session identifier and places it on the user's hard drive in the form of a "cookie", a small bit of unique code which binds the user to the server and makes the session persistent, or "sticky". The dispatcher takes into account the importance of the cookie and routes all requests for that session to the same analysis server.

SCALING WITH THE CLUSTERED ALPHABLOX ANALYSIS SERVER

As a 100% Web-based infrastructure for analytic applications, Alphablox has implemented this scalability solution with the clustered Alphablox Analysis Server, and has certified Resonate's Central Dispatch traffic scheduling product for load balancing across the analysis servers.

With this solution, analytic applications enjoy higher standards of performance, as the analytic infrastructure distributes the demands we explored in "Scalability—The Business Perspective" across multiple servers. Similarly, this solution addresses the issues we saw in "Scalability—The Technical Perspective" by preserving the user's experience with the resources available to the IT team. With performance and scalability bottlenecks removed, the organization can rely on its analytic infrastructure to help pinpoint the best opportunities to manage its business profitably from the front lines, rather than rely on its managers to gather and interpret data.

Resonate's Central Dispatch provides a good match to Alphablox's Web-centric focus. Its load-balancing solution is implemented in software, offering flexibility. By residing inside the firewall it preserves security, and offers the organization scalability without the encumbrance and expense of yet another hardware decision to make and yet another machine to maintain. Its support for cookie-persistence is critical to Alphablox applications, which use cookies to move session identifiers between the Alphablox Analysis Server and clients.

CONCLUSION

For each of the business and technical issues explored in this white paper, the clustered Alphablox Analysis Server implemented with Resonate's Central Dispatch load-balancing product has a compelling feature and benefit. Business managers and IT teams seeking scalability of analysis servers will find in this combined implementation the stability, flexibility and performance required to keep pace with the evolution of the analytic infrastructure. As this infrastructure evolves, so grows the ability of decision-makers at the front lines to immediately pursue their most profitable courses of action—so that they can survive and win under changing business conditions.

For more information on the clustered Alphablox Analysis Server, contact Alphablox:

Tel:	1.888.BLOX.NOW (1.888.256.9669)
Email:	info@alphablox.com
Web:	www.alphablox.com
Worldwide Headquarters	520 Logue Avenue Mountain View, CA 94043 Tel: 650.526.1700
Alphablox Corporation Limited	1 Farnham Road Guildford, Surrey GU2 4RG United Kingdom Tel: +44.1483.549.029
Alphablox Australasia	266 Auburn Road Hawthorn, VIC 3122 Australia Tel: +61.3.9810.6300