zapthink white paper

SOLVING THE VERY LARGE MESSAGING PROBLEM IN THE ENTERPRISE

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

SOLVING THE VERY LARGE MESSAGING PROBLEM

February 2005
Analyst: Ronald Schmelzer

Abstract

Companies are increasingly seeking to tie together their disparate enterprise using the promising, but emerging technologies of XML, Web Services, and Service-Oriented Architectures. These approaches promise significant business agility in the face of IT heterogeneity. However, these benefits come at a price: performance and efficiency. As the network traffic increases due to the increasing size and volume of messages, both XML and non-XML based, existing corporate IT infrastructure will be taxed to its limit. General-purpose application servers, network equipment, and messaging infrastructure will be increasingly devoted to simple message parsing, handling, and routing functions, while precious few resources will be left to execute the core business logic so important to companies.

Research shows that the quantity and size of these metadata-laden messages won't be decreasing soon. Developers and specifications bodies continue to tax messaging systems with additional layers of headers and metadata meant to abstract underlying infrastructure. Increasingly large message size, along with a general increase in message volume, combine to create the challenge of Very Large Messaging (VLM).

Previous approaches to solving distributed messaging problems, including messaging middleware, ESB, and application servers, were not designed to handle the challenges of VLM. Emerging approaches such as hardware appliances and binary XML, may solve part of the overall VLM problem, but fail to provide a comprehensive approach that targets all systems, networks, and processing infrastructure that runs within the corporate IT environment. Further, while these approaches may remove some of the overhead of message parsing, they offer no direct benefit to assimilating and utilitizing these messages within applications.

As a result, new approaches are needed to deal with messages being exchanged on the network that are exceeding the capabilities of the general purpose hardware and software that is now being applied to the problem. How can efficient content-level message processing be distributed to all the nodes in a corporate network? How can dumb networks be made more intelligent through the ability to process data and metadata it formerly ignored? In this paper, approaches to the "Very Large Messaging" (VLM) problem, and potential optimal solutions that hope to break the stalemate in network processing of data are presented.

Table of Contents

I. The Challenge of Very Large Messaging 4
The Growth of XML and VLM Traffic on the Network 5
II. Approaches to Solving VLM Processing Challenges 8
Network appliances / hardware approaches 8
Grid and Blade-based approaches 9
Emerging Chip-based solutions 10
III. Optimizing Very Large Messaging in the Enterprise 11
IV. Conclusions 12

I. The Challenge of Very Large Messaging

The nature and quantity of traffic on the network continues to expand unabated as technologies like email, Web, and business-to-business networking have become ubiquitous in the daily lives of business users. In addition to these well-established technologies, an emerging class of interoperable, standards-based mechanisms for connecting applications and organizations together are starting to emerge. The interoperability benefits of XML and Web Services, the significant integration cost savings that Web Services provide, and the business agility that Service-Oriented Architecture (SOA) promises are gaining significant currency in today's enterprises.

Because this new kind of traffic is content-oriented, rather than protocol-oriented, solutions responsible for routing, managing, securing, processing and utilizing XML traffic must make decisions based upon the content of the messages, rather than the protocols that underlie those messages. In addition, XML is metadata-rich, meaning that it not only encodes the data that is exchanged between applications and/or businesses, but also encodes the meaning of that data and additional information meant to assist in proper processing of the information contained in a message. More importantly, the point is that after you get, parse, validate the message, you need to **do** something with it that is useful to the business – and this is where the majority of the processing effort should take place.

As a result, formats like XML generate significant bandwidth, processing, and storage requirements. XML processing tasks such as document transformation, parsing, message validation, classification, security, and intelligent routing are inherently processing-intensive, placing a significant burden on server infrastructure that may not be optimized to perform these tasks.

At the same time, enterprises are looking to add more layers of functionality, and thus complexity, to the XML traffic on the network. Simple point-to-point XML-based exchange in free-text format is no longer sufficient to realize the benefits that XML promises. Companies increasingly desire a set of requirements for XML-aware network processing such as routing, transformation, compression, caching, security, and management tasks. As a result, as network traffic based on XML increases, IT data center administrators and developers are quickly realizing that the operational inefficiencies of XML are bogging down their general-purpose hardware and software. The addition of more advanced security, reliability, and process capabilities puts an overwhelming burden on existing network infrastructure that is already stretched to the limit handling basic XML processing tasks.

The Emergence of the SOA Performance Crisis

Text-based, metadata-laden XML is intended for both machine processing and human readability. The combination of these two purposes results in message sizes that are easily 10 to 50 times larger than corresponding messages written in binary protocols or flat-text formats. XML-conversant endpoints must perform the steps to receive, decrypt, validate, parse, transform, perform business logic, serialize, canonicalize, sign, encrypt, and transmit on a per-message basis, imposing a significant load on processing machines. Research has shown that basic XML tasks such as conversion of data to a canonical format represented over 93% of the total processing time for processing of simple documents like an XML-Signature document (a typical XML document).

Add to these processing requirements the need for additional parsing and validation steps for XML schema, the growing number and complexity of security, reliability, and process headers, the need for partial-message security, XPath and XQuery processing of documents, message compression and decompression, business logic validity checking, and message integrity validation and a bandwidth, processing, and storage nightmare results that grows in cost and complexity over time. Clearly, general-purpose processors, off-the-shelf software parsers and validation engines, general application servers, and non-optimized security solutions are not sufficient to meet the burgeoning XML processing challenge.

Another reason for the impending SOA performance crisis is that while the technology that powers SOA has been evolving for the past few decades, the best practices that guide their usage are incredibly new and emerging for most developers. As a result, ZapThink has seen some particularly egregious development and architectural practices that put an inordinate amount of extra traffic on the network, especially as developers seek to replace proprietary interfaces with XML standards-based ones. Until best practices are learned, network traffic will continue to grow unabated.

Even when proper best practices are applied, SOA mandates the use of coarse-grained, business-oriented Services. That is, it mandates using a smaller number of more highly reusable and composable Services to replace a larger number of fine-grained Service calls. While this may result in fewer interactions on the network, coarse-grained interactions can increase the size of the overall payload, resulting in a fewer number of larger-sized XML documents.

Finally, an ever-increasing number of new vertical industry and domain-specific languages are emerging to solve the needs of a wide range of users and constituencies in the market. These standards continue to grow as end-users make their requirements known to the standards bodies. As companies accept and adopt the various different specifications currently being proposed, the size of the overall messages on the network will increase. Thus, standards-based, Service-oriented, coarse-grained, XML-based messages will continue to grow in size over time. All of the above factors – the inefficiency of XML, lack of best practices by developers, composite nature of SOA, and proliferation of industry standards will result in a performance crisis for SOA implementations. Clearly, this problem will not solve itself or disappear on its own. Rather, companies now will face the requirement of dealing with Very Large Messaging (VLM) in their IT infrastructure. The VLM problem encompasses both large messages as well as a high volume of smaller messages – the combination of both factors will cause pain to companies implementing SOA.

However, the presence of XML itself on the network isn't sufficient to cause alarm for network administrators and application developers. Rather, concurrent with the growth of metadata-rich XML traffic is the growth of the size of the messages themselves on the network. In many cases, the high impact of XML on the network simply can't be avoided because the XML payload itself is very large. Businesses typically engage in Very Large Messaging (VLM) with other organizations in addition to consuming them within the corporate network. As a result, they typically represent business documents critical for business-to-business interchange.

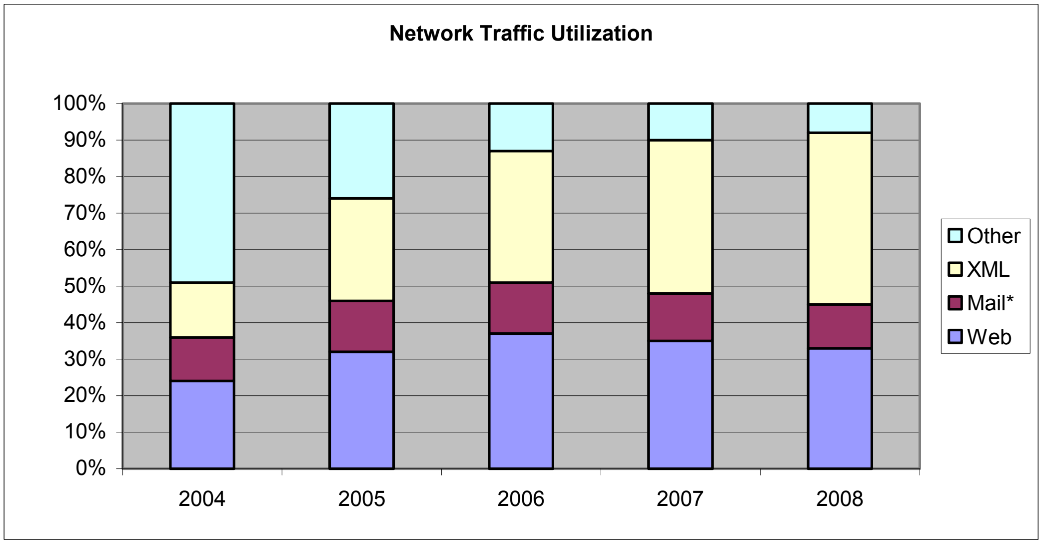
The Growth of XML and VLM Traffic on the Network

In addition to increased processor load, as XML usage expands in the network, inefficient XML traffic will increasingly consume network bandwidth. At some point, IT administrators will demand more effective use of network resources. ZapThink research shows that the average quantity and size of these metadata-laden messages will continue to increase over time.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

ZapThink expects XML traffic on the network to greatly increase over the next few years, from under 15% of all network traffic on the network in 2004 to around 48% of all LAN network traffic by 2008. Figure 1 and X below show the expansion of network traffic over the next few years:

Figure 1: Growth of XML Traffic on Network



Copyright © 2004 ZapThink LLC

Table 1: Traffic by Payload Type

	2002	2003	2004	2005	2006	2007	2008
Web	11%	18%	24%	32%	37%	35%	33%
Mail*	8%	10%	12%	14%	14%	13%	12%
XML	2%	4%	15%	28%	36%	42%	47%
Other	79%	68%	49%	26%	13%	10%	8%

(* by Mail we mean SMTP, POP, and IMAP traffic for mail that is not HTTP-based)

While close to 50% of all network traffic by 2008 will be XML-based, the purpose and type of XML traffic can vary widely. For example, some XML-based traffic is for Web Services for system-to-system communication mainly behind the corporate firewall to simplify integration, while other traffic might be XML payloads destined for outside the corporate network. In general, ZapThink sees the following major XML workload types as being prevalent on the network for the next 3-5 years:

- XML Web Services messages (SOAP-based)
- Representational State Transfer (REST)-based Web Services interchanges (non-SOAP based) – REST is a more simplistic way of implementing XML-based system-to-system exchanges that doesn't mandate security, management, process, or other overhead, but is implemented in a mostly synchronous fashion.
- XML document formats (such as ACORD messages, etc.) not transmitted via Web Services or REST-based mechanisms.
- XML-based device-to-device interchanges

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

ZapThink has attempted to categorize the various potential uses for XML that may be present on the corporate network and their growth from 2002 through 2008 as follows:

Figure 2: Makeup of XML Traffic on the Corporate Network

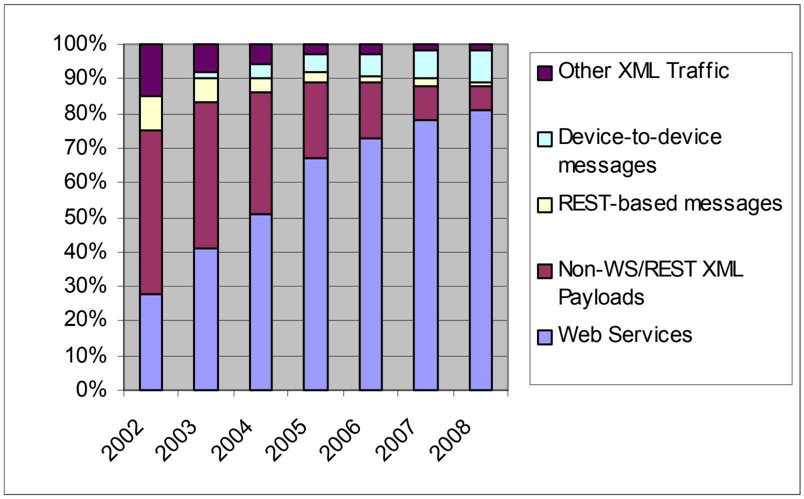


Table 2: Makeup of XML Traffic on the Corporate Network

	2002	2003	2004	2005	2006	2007	2008
Web Services	28%	41%	51%	67%	73%	78%	81%
REST-based messages	10%	7%	4%	3%	2%	2%	1%
Non-Web Services or REST XML Payloads	47%	42%	35%	22%	16%	10%	7%
Device-to-device	0%	2%	4%	5%	6%	8%	9%
Other XML Traffic	15%	8%	6%	3%	3%	2%	2%

Given the above predictions and estimates for future traffic on the corporate LAN, successfully addressing the challenge of managing the efficiency of Web Services traffic in particular will be key to assuring the viability of XML as a robust and reliable protocol on the network.

Generally speaking, a message can be considered to be “Very Large” if it is roughly 1 megabyte in size or if it contains highly repeated XML elements and very deep levels of nesting. Over time, as XML infrastructure is increasingly optimized, what will be considered to be “very large” will also shift over time. By the year 2008, a very large message will be 100 Megabytes in size or larger. The below table shows the percentage of messages on the network that will be VLM, and the growth of average VLM size over the next few years:

Table 3: Growth of VLM Traffic on the Network

	2004	2005	2006	2007	2008
Average size of VLM traffic (Mbps)	1	3	10	30	100
VLM as % of XML Traffic	3	8	12	18	25
XML Processing speed for Avg. Network (Mbps/Sec)	1	10	50	100	500

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

It is clear that while XML processing performance will steadily increase over the next few years, so too will the average size of XML documents on the network. And raw processing is not enough – applications need to do something with the messages they receive, often as part of complex multi-step processes As a result, the VLM problem will not be solved simply by applying more hardware resources to the problem, but rather by new approaches to optimize XML handling on the network.

II. Approaches to Solving VLM Processing Challenges

Network traffic increases due to both the increasing quantity and size of messages, both XML and non-XML based, will tax existing corporate IT infrastructure to its limit. Network administrators will increasingly devote general-purpose application servers, network equipment, and messaging infrastructure to simple message parsing, handling, and routing functions, while precious few resources will be left to execute the core business logic so important to companies. As a result, there is a need for approaches that seek to provide the content-level functionality required of today's XML and Web Services solutions but also provide the high level of performance needed to effectively run these solutions in production.

Consequently, companies have applied a variety of approaches to deal with messages on the network that stress the capabilities of the general purpose hardware and software that now deals with the problem. Some of these approaches are hardware and software approaches that aim to provide high performance processing of XML and mitigate the threat posed by Very Large Messaging on the network by either processing them with a very high degree of efficiency or by limiting their presence on the network.

Network appliances / hardware approaches

While there are many different approaches to providing high performance XML solutions, the hardware solution remains a good option for IT data center administrators looking to manage corporate-wide XML and Web Services traffic. The hardware form factor, whether specialized chipsets and custom-developed firmware, or specialized, pre-configured software on more general-purpose hardware platforms, offers some significant benefits to customers looking to improve the performance of their XML solutions. Many hardware platform vendors are able to offer substantially higher performance over purely software-based solutions, and also offer security-hardened environments that prevent tampering as well as simplified administration. Hardware solutions also free application developers from being responsible for protecting their applications from every possible type of attack by offloading those responsibilities onto a centralized device, and focus them on the primary responsibility of building application logic.

IT personnel can configure hardware appliances so that they are ready to install by simply plugging the equipment into their corporate network with minimal additional configuration. As a result, IT shops can control their installation and maintenance costs and complexity, making sure that they have properly configured certain security, routing, transformation, and management features prior to installation.

Hardware appliance solutions face the challenge of making sure their solutions have a place in existing IT administration environments. The glut of Network Appliances and devices over the past five years has resulted in a complex "stack" of rack-mountable appliances covering everything from Voice-over-IP applications to search appliances. Even if there is physically room in the stack, there simply may be no interest of IT administrators to add yet another device to the stack, or to change their network topology to suit the new device. Data center administrators want to see a consolidation of devices rather than addition of new, untested applications. Clearly, there is some clearing out of rack space going on currently, with a movement to more space and resource-efficient server "blade" configurations. As this process progresses, there will be more room for hardware appliance solutions, but in some cases, the lack of rack space can pose challenges. In addition, there is the broader problem of hardware total cost of ownership (TCO). TCO includes buying and housing the hardware, but more importantly it includes the people to take care of it – and this is not inconsequential for more firms.

Grid and Clustering approaches

Companies are also looking to solve their performance problems by grouping their computing resources together into purpose-built computational clusters, or more general-purpose computing "grids" that aim to solve the general performance requirements of the company. The grid-based approaches are gaining significant traction within companies as the cost and availability of commodity infrastructure, such as application servers, operating systems, and even computing hardware reaches a point where there is sufficient capability and capacity on the network to handle application-specific processing needs. The Grid approach to highly scalable, componentized, and interchangeable computing power is especially taking hold in the Financial Services industry where compute tasks are frequently repeatedly executed over the course of the business day.

While Grid-based approaches offer a compelling value proposition for instances of centralized processing of XML and Web Services traffic as they pass from one point on the network to another, they don't address the challenge of XML processing once the traffic reaches its end destination. Furthermore, simply adding computing power through brute force doesn't solve the issue of the burgeoning network traffic glut and developer inefficiency. If anything, Grid-based systems would benefit from a more optimal and efficient approach to VLM handling overall.

Blade-based approaches

Another hardware form factor rapidly gaining acceptance is the blade form factor, increasingly used in emerging virtualized environments for high-performance computing. A blade server is actually a group of individual, general-purpose, commodity servers pulled together into a single appliance called a chassis that provides the group of servers with a common set of infrastructure like power supplies, cooling fans, and networking capabilities. Blade computing seeks to reduce IT complexity and administration costs by simplifying the management of groups of servers, speed deployment by allowing the swapping of individual servers at very low cost and with significant ease, conserving space and power by consolidating servers into small form-factor chassis, and providing for significant agility by allowing the provisioning of servers at runtime.

A blade server consists of general-purpose computing hardware adapted with specialized software to perform specific processing tasks, instead of purpose-built hardware like network appliances. For this reason, companies that are looking to solve XML performance issues are increasingly looking to the blade form factor to deploy their solutions. In particular, the embeddable software approach enables companies to deploy their solution on a flexible, as-needed

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

basis to available blade servers as XML performance requirement demands change.

The advantages of the blade form factor is that it doesn't require any changes to network topology and can make use of the management and runtime infrastructure of existing blade systems. Some disadvantages of the approach include the fact that chassis are far from standardized, requiring technology vendors to carefully select their partners, and a requirement to involve the data center administrator in the purchasing process.

Binary XML

One emerging approach for improving the performance of XML processing treats XML as a binary format – rather than the text-based format so often maligned as the root of XML's inefficiency. Such an iconoclastic approach to XML flies in the face of the conventional wisdom about the benefits of text-based XML. Nevertheless, binary XML is gaining some traction in the marketplace, and may help solve many of the performance problems that promise to swamp tomorrow's XML-laden networks.

To resolve the limitations of all-or-nothing compression and its processing overhead, the W3C has begun the development an alternate, binary encoding of XML that promises to significantly alter the processing, bandwidth, and storage penalties that currently plague XML. This encoding uses binary, rather than text-based, means for serializing and transmitting XML information. This binary representation of XML is far more sophisticated than simply compressing the XML format into a binary form. The binary XML approach takes advantage of XML language grammar to simultaneously compress, validate, and optimize the processing of XML documents.

Since XML is a text-based format, using common binary compression formats like zip can squeeze over 90% of the volume out of XML files and documents. However, the problem with compression is that it actually increases the amount of processing required before transmitting an XML document and again before parsing it at the receiving end. So, compression may solve the bandwidth issue, but it worsens the processing problem. In addition, GZIP is not type-aware and does not compress large sets of floating-point numbers well. Furthermore, compression and encoding formats like zip and base64 offer an "all or nothing" approach – once a message is encoded, the recipient must decode the entire message in order to work with any part of it. However, much like compression, usage of base64 or equivalents requires a first-pass encoding step as well as a decoding step once the document is received by the end-point, so any marginal gains in network bandwidth are then lost in processing time. Finally, it's quite likely that one might not even realize network performance gains using encoding, since encoded documents can often be much larger than their original format.

Binary XML, however, is not without its downside. The greatest challenge of any binary encoding is that all points on the communication path need to be able to not only tolerate the format, but be able to process it. While proponents often talk about how end-points can easily be configured to deal with binary XML, they often neglect the fact that intermediaries between the communicating parties often must be able to inspect and make decisions regarding that traffic. As a result, binary XML's global acceptance hinges upon all security, process, management, and transformation systems or devices being able to understand and process the binary XML format. In addition, the standards for Binary XML have yet to be widely accepted or adopted. Furthermore, binary XML raises the specter of potential compatibility and vendor lock-in concerns. For example, the format chosen to represent numerical data, such as integers, floating point numbers, or arrays, must be platform independent, so that different consuming platforms are able to take advantage of the performance boost that such native formatting offers – a tall order in today's complex, heterogeneous IT environment.

It is also not clear if solving the parsing and transmission problems of XML will truly result in significant overall performance increase. In many situations, XML processing represents only a small part of the overall processing load for a given XML message. Many of the proposed gains would be achieved by skipping aspects of XML processing that many consider essential. Finally, Binary XML does not address the processing costs that result from security look-ups, semantic mapping, transformation, and other complex processing tasks, suggesting that binary XML might not be worth the trouble, since the processing bottleneck may be elsewhere.

Emerging Chip-based solutions

In addition to the network appliance, blade, and PCI-card form factors, an emerging set of vendors is producing specific chipsets for handling XML processing requirements. Those chips are either produced as Field-Programmable Gate Array (FPGA) units that have a high per-unit cost but allow for faster time-to-market or as application-specific integrated circuits (ASICs) that can be produced at high volumes at low cost but have a higher capital investment and longer and more costly development cycles. Regardless of the approach used to develop and produce the technology, companies are aiming these chips at the emerging market for third-party software and hardware vendors looking to embed high-performance XML processing capabilities in their systems by utilizing specialized capabilities of a chip rather than over-burdening their generic CPU.

While each of the solutions mentioned above contribute to improving XML and VLM processing on the network, many leave much desired as far as the ability to cover all message processing nodes on the network – clearly hardware solutions alone aren't enough to successfully address the VLM problem on the network as it grows over the next few years.

III. Optimizing Very Large Messaging in the Enterprise

While hardware in its many forms offers compelling solutions for optimized XML processing, such solutions aren't the only approaches to improving XML performance on the network. Clearly, one of the first approaches companies should take is to reevaluate the usage of generic parsing and processing technologies from their vendor suppliers. Rather than code their own security, transformation, routing, or parsing routines, companies should seek more optimized software solutions in those cases when a hardware approach is not viable or cost-effective, such as on end-point processing nodes, sometimes-connected devices, mainframes, and other systems. In these cases, a variety of software-based approaches are emerging for more optimized handling of XML performance.

Software optimization:

- Avoiding repeated parsing / optimized parsing – Developers must begin to curtail emerging bad habits and practices around repeatedly encoding, serializing, and parsing of XML documents. Software approaches that optimize the number of steps required to get

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

information into and out of XML document form will be highly sought-after.

- Optimize to OS/hardware (native code) – Another optimized software approach is to tailor the specific, compute-intensive requirements for XML parsing to the specific operating system or hardware platform that is executing the code. To be most useful, these optimizations should happen without the explicit knowledge or interaction of the developer.
- Eliminate canonical format conversion – Too many transformations and canonical representations of XML can result in performance degradation. The elimination of unnecessary steps can go a long way towards optimizing XML performance and reducing the VLM problem.
- Intelligent scaling (service grids) – Finally, the use of commodity infrastructure components to allow for dynamic, and cost-effective scaling of processing is quite compelling in a scenario where VLM is an increasing threat to the health and well-being of the network.

Some vendors are improving the performance of XML processing tasks without focusing on a specific application domain such as security, transformation, or routing. These vendors solve the core problems that bog down most processors when they encounter XML traffic on the network. In particular, the use of either specialized chipsets or software can significantly accelerate and optimize parsing, validation, document query, and compression steps by offloading those tasks from general purpose computing infrastructure.

The clear advantage of a software-based approach is that performance-enhancement to address VLM processing challenges can be implemented at the end-points in general purpose hardware infrastructure, rather than through hardware proxies sitting between communicating end-points, or on specialized hardware products. The advantages are that the burden of processing XML and other content-intensive, metadata-rich messaging formats can be distributed on the network and not centralized on overburdened infrastructure.

IV. Conclusions

As companies seek to increasingly make use of secure, reliable, process-driven, and loosely coupled Service-oriented Architectures built on Web Services, they will quickly come to realize what sort of performance penalty imposed by the inefficient XML format. Only optimized approaches to handling XML will alleviate the problems companies face in utilizing this potent new technology. In addition, the performance challenges imposed by Very Large Messaging will threaten the viability of SOA implementations, and could precipitate an impending SOA performance crisis.

Hardware solutions alone are not enough to deal with the performance challenges introduced by Very Large Messaging in the network. Companies require a distributed approach that leverages general purpose computing infrastructure, in addition to optimized networks, to effectively tackle the key performance issues. In addition, a new generation of efficient software products is required to serve the growing needs of the modern IT environment.

ZapThink sees a significant uptake in Service-oriented application development during 2005. But, as with any new technology paradigm, early implementers will run into serious difficulties. Companies will experiment with SOA applications, finding that they need to reconsider designs for acceptable results. We see the growth of Very Large Messaging (both size and volume of service-based

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

messages) choking application performance in some cases, causing developers to "go back to the drawing board." However, even with these growing pains, we expect to see SOA as an inevitability in the industry by year's end.

The evolution to SOA and general pressure on IT cost efficiencies and demand for results will drive new ways of managing and deploying applications. Most notably is the convergence of inexpensive blade systems and distributed software capabilities into grid computing solutions. So far, many companies have experimented with grid for utility computing problems, such as long running compute-intensive processes, but the evolution of grid technology for real-time business transactions has yet to be realized.

Mounting pressures from open source offerings and customer demands will force hardware and software vendors to look at new models for delivering and pricing their wares. These challenges open up new opportunities for companies that are nimble and flexible in evolving new business models to take advantage of the ever-changing IT environment as it moves down the inevitable path towards SOA.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Copyright, Trademark Notice, and Statement of Opinion

About ZapThink, LLC

ZapThink is an IT market intelligence firm that provides trusted advice and critical insight into XML, Web Services, and Service Orientation. We provide our target audience of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing—a vision of IT meeting the needs of the agile business.

ZapThink's role is to help companies understand these IT products and services in the context of SOAs and the vision of Service Orientation. ZapThink provides market intelligence to IT vendors who offer XML and Web Services-based products to help them understand their competitive landscape and how to communicate their value proposition to their customers within the context of Service Orientation, and lay out their product roadmaps for the coming wave of Service Orientation. ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into how to assemble the available products and services into a coherent roadmap to Service Orientation. Finally, ZapThink provides demand intelligence to IT vendors and service providers who must understand the needs of IT users as they follow the roadmap to Service Orientation.

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOAs by vendors, end-users, and the press. They are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted industry analysts in the IT industry.

ZapThink was founded in October 2000 and is headquartered in Waltham, Massachusetts. Its customers include Global 1000 firms, public sector organizations around the world, and many emerging businesses. ZapThink Analysts have years of experience in IT as well as research and analysis. Its analysts have previously been with such firms as IDC and ChannelWave, and have sat on the working group committees for standards bodies such as RosettaNet, UDDI, CPExchange, ebXML, EIDX, and CompTIA.

Call, email, or visit the ZapThink Web site to learn more about how ZapThink can help you to better understand how XML and Web Services impact your business or organization.

ZAPTHINK CONTACT:
ZapThink, LLC
11 Willow Street, Suite 200
Waltham, MA 02453
Phone: +1 (781) 207 0203
Fax: +1 (786) 524 3186
info@zapthink.com

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com