Java™ Computing in the Enterprise

What It Means for the General Manager and CIO

Bud Tribble

With Contributions by
Vicki Morris, Thomas Schroeder and Charles Tierney

Sun microsystems

THE NETWORK IS THE COMPUTER™

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043 USA
415 960-1300 fax 415 969-9131

ABOUT THE AUTHOR

Guy “Bud” Tribble is vice president and chief architect of Java systems at Sun Microsystems Computer Company. He is one of the industry’s noted experts in object-oriented programming and user interface design. For example, he was key architect of the NEXTSTEP operating system (as a co-founder of NeXT Computer). Prior to that, at Apple he was head of the entire Macintosh software engineering effort, including software architecture, user interface design and applications. Tribble holds a BA in biophysics and an M.D. and Ph.D. in neurophysiology.

Contents

Overview 1

Sun's Java™ technology has been described as a "revolution" in computing. The essence of this revolution is a shift in point of view from a "desktop-centric" model of computing to a "network-centric" paradigm. An important implication for the enterprise of this change is to move complexity off the desktop and onto the network and servers, where it can be centrally and professionally managed. "Java Computing" does this while continuing to deliver as much or more of the local processing power so important for today's graphic and multimedia-intensive applications.

There is much discontent in today's enterprise computing, stemming from several causes:

- It is too complex
- The constant administration and upgrading of desktops is too expensive
- It is not secure enough
- It is not reliable enough
- Incompatible desktops prevent universal access to applications
- Applications take too long to develop and deploy

Java Computing addresses all of these problems. Just as important, its platform independence makes Java a paradigm that can be adopted incrementally, without making obsolete existing investments in hardware and software. Furthermore, a user can start today for far less investment than it takes to move to the next revision of the Windows/Intel architecture.

The Internet (and intranets) and Java have launched a huge wave of activity in the computing industry. All leading OS and browser platforms are building Java into their systems and major development tool vendors and application providers are supporting it. Java “applets” are popping up all over the Web, bringing new live capabilities to a formerly static medium. And a growing number of Global Fortune 1000 corporations have embraced Java networking as a facet of their mission-critical enterprise computing systems.

Much has been written about Java technically, but little has been said about what Java and other Internet technologies mean to a corporate executive trying to run his or her business. This paper is intended to address three questions:

1. What is Java Computing and what does it mean for enterprise computing?
2. Why does Java Computing make business sense?
3. How does a CIO develop, deploy and manage Java Computing in the enterprise?

This paper concludes that Java and other Internet technologies can offer a major improvement in simplicity, expense, security and reliability versus some enterprise computing environments in place today.

Based on Sun's internal experience with network computing, it is estimated that the full annual cost of operating Java-based clients in an enterprise environment will be less than $2,500 per seat. (“Java-based clients” are referred to in this document as “thin clients.”) Typical expenses in today's existing environments to manage heterogeneous desktop-centric clients or “fat clients” (e.g., WindowsNT, Win95, UNIX, OS/2, Mac, etc.) are in the $10,000 to $15,000 range.

For example, according to the Gartner Group, the annual cost of running such clients is estimated at $11,900 per seat. (Gartner Group, Strategic Planning Research Note SPA-140-22, April 26, 1996). It is suggested that these savings could be realized for as many as 90% of an enterprise's clients, with only the highly intense, highly diverse application users needing to continue with a fat client. Thus, savings from moving from a fat-client to a thin-client architecture could be as much as $84 million annually for an enterprise with 10,000 clients.

Besides saving costs, Java can ensure that necessary features and functions are available to all enterprise users, with application front-ends being accessed through a common “webtop” environment. “Webtop” refers to a desktop environment that can run Java applications. This may be either a Java-enabled browser on an existing desktop platform, or a Java device desktop. Most significantly, nothing that exists today in current enterprise computing environments need be discarded to begin realizing these benefits; i.e., Java applications can be enabled on currently installed computing platforms, followed by a phased migration to the most economic platforms. Java-based client front-ends can likewise be added to existing legacy backends in a phased, incremental approach.

In the same way that existing enterprise information can be quickly retargeted and deployed to the intranet Web, existing enterprise applications can be quickly retargeted to the webtop. Just as departments can customize their own Web pages, rich sets of Java application building tools can facilitate decentralized customization of application front-ends.

A big advantage of the Java Computing model is that users can log on anywhere they have access to the intranet Web and have full access to their complete home environment. Administrative costs for moving users are greatly reduced, and dial-up access at home or while traveling can give users the same full access to their data and applications. Connected mobile computing is thus greatly enhanced.

Another force has brought Java to the forefront of computing. The Internet is becoming a new channel to reach customers, suppliers and business partners, and the World Wide Web allows the enterprise to selectively provide information access via the Internet. Java expands this capability, enabling companies to offer selective access across the firewall to transactional enterprise applications. The ability to give customers, suppliers and partners this Java-based window into the enterprise for delivery of goods, services and information flows seamlessly from the adoption of a Java-based enterprise computing architecture -- also called Java Computing. This adds a top-line management motivation for Java enterprise computing to the already compelling cost-saving benefits mentioned above.

These advantages have combined to make Java the de-facto standard for network-based computing on the Internet and intranet. Nearly all of the software industry has licensed Java, has committed to use it or is performing extensive evaluations of it. Forrester Research reports that of the Fortune 1000 firms it surveyed, 62% already use Java for some development and more than 40% expect Java to play a strategic role in their companies within the year. (The Forrester Report: Software Strategy Service, May, 1996)

It is suggested that before an enterprise invests additional millions in upgrading to new fat clients (e.g., upgrading clients to WindowsNT), a serious review be made of economic, security, reliability and feature/function benefits that can be realized today by migrating applications to Java and computing platforms to Java devices.

The up-front costs of moving to Java-based computing will be less than the cost of another upgrade of fat clients. The ongoing cost of Java-based enterprise computing could be less than a fifth of the cost per seat of a fat-client architecture.

To begin this migration process, the following steps are proposed:

1. Today:	Build an enterprise intranet infrastructure using Internet technologies.
2. Today:	Deploy a “webtop” (Java-capable browser) on all enterprise desktops.
3. Today:	Move electronic mail to the intranet with access through the “webtop.”
4. Today:	Identify “dedicated-use” applications -- desktops where employees run a small set of custom applications all day long as their job. Write or re-write these applications in Java.
5. In 6 Months:	Begin deploying Java devices in these dedicated-use areas.
6. In 12 Months:	Begin to migrate other applications to Java so they can be accessed through the webtop.

7. **In 18 Months:** When a critical mass of Java applications is present, begin replacing fat-client desktops with Java devices.

A detailed list of recommended next steps for enterprise general managers and CIOs is included at the end of this paper.

What is an Intranet? 2

A prerequisite to Java Computing is the deployment of an enterprise intranet infrastructure. The term “intranet” is widely used to describe the application of Internet technologies in internal corporate networks. Generally, businesses use intranets today to more effectively share information, taking advantage of the Internet browser paradigm. Intranets are based on technology that already exists in many enterprises.

Intranets provide many benefits to enterprises. Use of the browser paradigm simplifies internal information management and improves internal communication. Similarly, Web navigation and search paradigms make it easier for people to find and analyze information. Integrating Internet technologies with an enterprise infrastructure and legacy systems enables a company to leverage its existing technology investments.

In the past, Web servers focused primarily on storing static pages viewable from a browser. The simple ability to transfer these pages using the HTTP protocol was the requirement of an Internet server. Now, Web pages are becoming applications -- Java applications. They are downloaded on demand from a Web server and are available on any machine able to access Web pages and execute Java -- both Java devices and existing desktop platforms running Java-enabled browsers.

The basic infrastructure for an intranet consists of an internal TCP/IP network connecting servers and desktops, which may or may not be connected to the Internet through a firewall. The intranet provides services to desktops via standard, open Internet protocols. In addition to TCP/IP for basic network communication, these also include protocols for:

• Browsing	HTTP
• File Service	NFS
• Mail Service	IMAP4/SMTP
• Naming Service	DNS/NIS+
• Directory Services	DNS/LDAP
• Booting Services	Bootp/DHCP
• Network Administration	SNMP
• Object Services	IIOP (CORBA)

These open Internet protocols are well established in the industry. Many products exist for their support and management. For example, all of these services are supported by the Sun™ Solaris™ operating system, and can be managed by Sun Solstice™ management products.

Intranets and the Internet

Since intranets and the Internet are so closely related, it is important to consider how they interact and how they are typically utilized. One way to categorize these networks is illustrated in Figure 1. This framework divides business use of networks between internal (intranet) and external (Internet), and between informative and transactional use.

Sun's experience indicates that companies typically traverse this framework in a counter-clockwise direction -- progressing to increasingly sophisticated use:

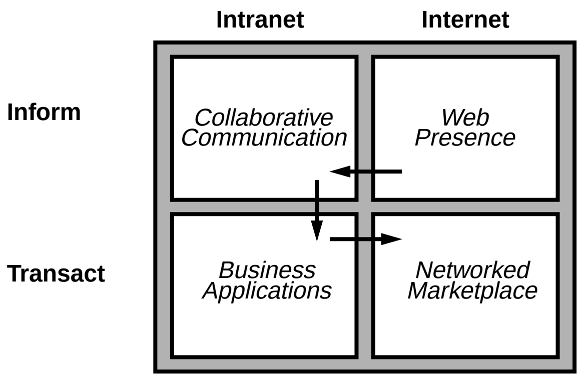


Figure 1 A framework for business networks

- *Web Presence*: Companies typically first use the Internet to establish a Web presence, informing customers about their products and services.
- *Collaborative Communication*: Just as a World Wide Web presence informs customers, an internal Web presence can be used to inform and to educate employees.
- *Business Applications*: With a set of internal Web sites established, the next step is to use Web servers as gateways to legacy systems, providing Web-based access to enterprise-critical data. This use may be transactional. Examples range from employee expense reporting to inventory control systems to data-mining applications.
- *Networked Marketplace*: The ultimate use of intranets occurs when internal business applications mature to the point where they can be released for customer/supplier interactions. At this point, the firewall can be opened up for specific activities and true electronic commerce enabled.

What is Java Computing? 3

The primary focus of this white paper is to describe the use and benefits of Java in enterprise computing environments. A short description of Java is presented here. For more, see “http://java.sun.com/doc/general.html.”

In brief, Java is an application development platform providing a portable, interpreted, high-performance, simple and object-oriented programming language and run-time environment.

Characteristics associated with Java include:

- Applications are much more secure than schemes running native code because the Java run-time system (part of the Java Virtual Machine) inspects all code for viruses and tampering before running it.
- Applications are adaptable to changing environments because users dynamically download application code from anywhere on the network.
- Shortened application development and deployment through code reuse, easier testing and rapid deployment via intranet.
- Portability across computing platforms because the Java Virtual Machine is available on all systems.
- Applications are fast because today's processors can provide efficient virtual machine execution. JIT (just-in-time) compilation and direct Java execution in silicon can deliver even more performance.
- Simple to learn and use with component-level object programming.
- Graphical applications and GUI functions are high performance because of Java's built-in multithreading.
- Robustness because the Java run-time system manages memory. (Memory management errors are a major bug source in traditional software.)

The essence of Java Computing is a client/server model in which Java application code is dynamically downloaded from server to client on-demand. In some cases, the applications are stored in cache on a hard disk at the client location and in others, they are stored only in DRAM. Since applications normally reside on the server and are delivered only as needed, all administration can focus centrally on the server, and users are assured of access to the latest application release level.

Java applications can run anywhere the Java Virtual Machine software is installed. Thus, they can run in any Java-enabled browser (e.g., Netscape Navigator or Internet Explorer). This is a key feature that allows a gradual migration to a simpler-to-manage client -- a Java device. This is a desktop client machine that is connected to the network and can download and run any Java application, but is free of the complexity and client administration needs of a traditional PC.

Rather than a traditional OS, the Java device contains a simple Java OS with a Java Virtual Machine. The Java OS and Java Virtual Machine can be stored at the client in flash ROM or can be booted from the network. Client data storage is done centrally on a file server or servers, and all client administration and configuration control is accomplished centrally.

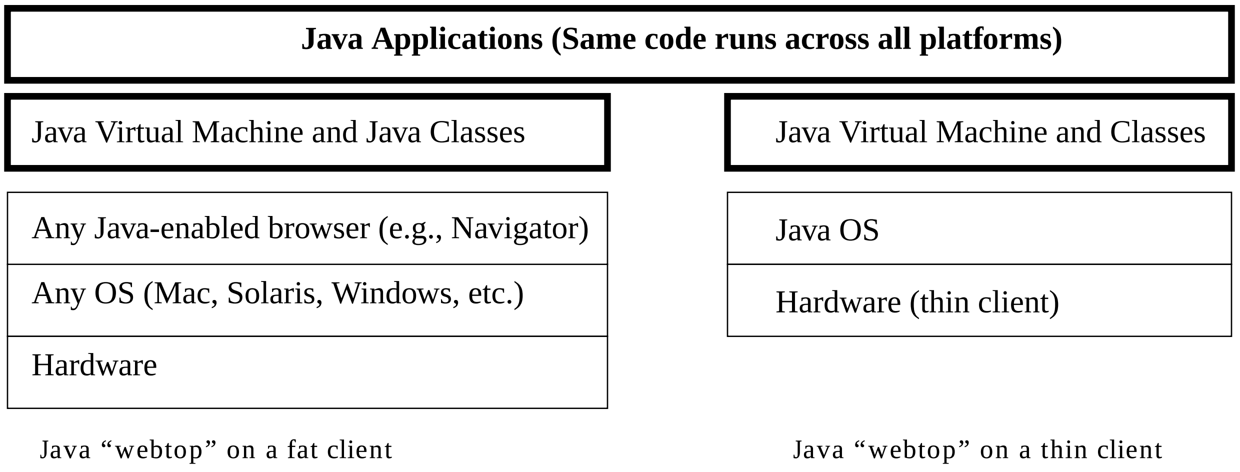

Java Applications (Same code runs across all platforms)
Java Virtual Machine and Java Classes
Any Java-enabled browser (e.g., Navigator)
Any OS (Mac, Solaris, Windows, etc.)
Hardware
Java Virtual Machine and Classes
Java OS
Hardware (thin client)
Java “webtop” on a fat client
Java “webtop” on a thin client

Figure 2 - Applications written in Java run on both fat clients as well as thin clients.

Applications running on the client communicate with servers via standard network protocols. A Java client may open a standard TCP/IP socket connection with an application server. Alternatively, more sophisticated client-server protocols can be employed. The JDBC (Java Database Connectivity) protocol provides SQL-oriented connectivity to databases. More complex, three-tiered applications can be built using distributed objects with industry-standard CORBA protocols. Three-tiered or multi-tiered application models are the key to leveraging existing back-end systems.

Since Java is truly platform independent, Java Computing spans much more than traditional desktop environments. Java clients will include "smart" telephones with built-in displays, PDA-like devices, set-top boxes, kiosks, point-of-sale devices, and even home video game machines, to name a few. Sun has licensed Java for use in all of these devices. They represent potential Java-enabled channels to customers and consumers for the services and interactions delivered beyond the firewall on the Internet. Java Computing gives companies an achitecture that leverages their enterprise computing investment across all of these emerging channels.

More technical information on Java can be found in "The Java Language Environment White Paper," by James Gosling and Henry McGilton, May 1996, available at "http://java.sun.com/doc/white_papers.html." Additional information for developers can be found at the "http://java.sun.com" web site.

Enterprise Computing Today 4

Some of the characteristics of today's enterprise computing environment include:

Infrastructure

- Fat clients with unnecessarily complex operating systems and incompatible interfaces.
- Servers in a variety of configurations, each networked in various ways to a subset of the clients.
- A variety of network topologies serving different subsets of the enterprise and having limited interoperability.
- Decreasing enterprise functionality and reliability as the complexity of the network infrastructure increases.

Applications

- A large body of legacy applications on a variety of platforms that cannot be immediately replaced.
- Server and client applications written in different languages that run on incompatible platforms and operating systems.
- Multiple -- and often incompatible -- client applications residing at various points on the network (e.g., incompatible word processing and spreadsheet packages).
- Multiple versions of network applications like e-mail that are either incompatible or do not interoperate properly.

Data

- Multiple and incompatible databases holding corporate data.
- Lack of a standard method to access corporate data uniformly, regardless of source.

Support

- Users who customize their local fat client with personally preferred applications that are not always compatible with network applications and may reduce enterprise reliability and / or security.
- Escalating support costs and a never-ending effort to keep clients and servers uniformly updated across the network.
- Large and dispersed support organizations dedicated to providing client on-site support.
- A possible migration to yet another version of Windows and WindowsNT that yields a near-zero return on investment.
- A big “Year 2000” investment challenge that yields a zero return on investment.

This chaotic environment, which spells a short tenure for many CIOs, is one that Java Computing is uniquely positioned to remedy. Java, together with the widespread adoption of intranet technology and standards, provides a way to simplify enterprise computing without scrapping existing investments.

What Problems Does Java Computing Address? 5

Many enterprise networks today are made up of DOS, Windows3.1, Windows95, UNIX, MacOS and WindowsNT clients networked in a variety of ways to a wide range of servers (often UNIX). There are several problems with this approach that have caused Java to become a catalyst for revolution. When the shortcomings of traditional architectures are combined with the existence of several interlocking computing environments (e.g., other client/server systems, mainframes, technical workstations and local PC applications), the result is exploding IT costs, with the general manager and CIO unable to obtain the information they need to run the business.

Problems addressed by Java Computing include:

Too Much Complexity

The complexity of fat clients loaded with up to one gigabyte of operating system and application files has led to high costs (up to $15,000 per year per seat when all costs are considered) and low reliability (potentially several reboots a day). The end user must spend a sizable fraction of his or her time as a system administrator as a result, lowering productivity.

Java simplifies things in a couple of ways. First, thin clients do not maintain any permanent state. All code, data and configuration information is stored and managed centrally. This combination of local desktop processing with central management drastically reduces the maintenance cost per seat without sacrificing the ability to customize. Applications are downloaded from the Web server on demand, so the user always has the latest version. Local administration of configuration files (e.g., AUTOEXEC.BAT, WIN.INI, SYSTEM.INI, PROTOCOL.INI) is not needed.

Second, the inclusion of the Java Virtual Machine on the server means that all clients and the server may be programmed as a single homogeneous platform -- the Java environment -- with consistent tools, training, and documentation.

Too Expensive

The detailed cost per seat varies across organizations based upon a number of decisions made and capabilities required by the enterprise. There appears to be some consensus about the range of costs for different enterprise computing architectures. The numbers below are intended to reflect fully loaded IT costs including the annual expense for client and server hardware and software, network infrastructure, application development and support, on-going user and dedicated maintenance resources, etc.

The table below summarizes ranges for the cost per seat in several architectures. These costs are in line with industry estimates of $10,000 to $15,000 per seat per year. For example, Gartner Group estimates the total cost per seat of PC platforms to be $11,900 per year (Gartner Group, Strategic Planning Research Note SPA-140-22, April 26, 1996).

Client Configuration Annual Full Cost per Year

- Heterogeneous, highly customized fat clients and servers (Wintel, UNIX, OS/2, Mac, MVS, etc.) with primarily local applications and local data file storage: $10,000-15,000 per seat per year.

- Homogeneous, standardized fat clients and servers (i.e., clients and servers are desktop-centric and binary-compatible -- Solaris clients and Solaris servers, for example): $9,000-12,000 per seat per year.
- X-terminal clients backed up by very-high-bandwidth network infrastructures and high-performance servers: $7,000-8,000 per seat per year.
- Diskless clients backed up by very-high-bandwidth network infrastructures and high-performance file servers: $6,000-7,000 per seat per year.
- Homogeneous dataless clients with all applications and data files stored on servers of the same architecture; e.g., Solaris servers with dataless Solaris clients that have a local hard disk only for virtual memory: $4,000-5,000 per seat per year. This data is based upon the experience at Sun of moving to a homogeneous dataless network architecture for its 35,000 internal-use network nodes.

The savings of moving from fat to dataless clients have already been captured internally by Sun.

The projected cost of dataless thin clients with all applications and data files stored on servers and generally executed on clients that include a hard disk for virtual memory (i.e., Java devices): less than $2,500 per seat per year. This estimate is based upon an analysis and extension of the cost elements of the homogeneous dataless environment that has already been implemented internally by Sun.

Appendix A shows a comparison of the economics of Java Computing vs. fat client computing.

Inadequate Security

Today's enterprise computing environments have a number of security weak points that are addressed by features in Java:

Strong Memory Protection - Java removes the possibility of either maliciously or inadvertently reading and/or corrupting memory locations outside boundaries of the program. This means Java applications or applets cannot gain unauthorized memory access to read or change contents.

Encryption and Signatures - Java supports the use of powerful encryption technology to verify that an applet came from an authorized source and has not been modified.

Rules Enforcement - Java is completely object based. By using Java objects and classes to represent corporate information entities, it is possible to state explicitly the rules governing the use of such objects. Since these rules are embedded within the objects themselves, introduction of ad-hoc access and manipulation methods is controlled.

Run-time Verification - Java also provides a run-time verification system that ensures that all applets downloaded to the client will not violate the integrity of the environment.

In addition, since Java devices get all of their programs and data over the network, they can be configured without local removable storage. In some situations, this is desirable in order to keep data within the network, or to keep viruses out of the network.

Inadequate Reliability

The complexity of current enterprise computing -- heterogeneous clients, servers, OSes, application and network topologies -- results in an inherent degradation of reliability. A failure at any point in these network interfaces can cause a failure in the application. Java, on the other hand, provides a seamless environment spanning the network, from server to client. The same Java applications run on all platforms and networks, and the simplicity of building the client and server software using the same Java programming platform significantly improves client/server reliability.

In addition, the Java language itself encourages the production of reliable, simple code. Since the language is object oriented, it promotes re-use, which, in turn, increases reliability. Java also promotes good software engineering practices with a clear separation of interfaces and implementations and easy exception handling. Java's automatic memory management and lack of pointers remove one of the largest causes of programming errors.

Applications Not Available to All Users

Because traditional enterprise environments require the porting of applications to each separate client environment, the cost of implementation on all clients cannot be justified for many applications. Furthermore, certain application features or functions may not be available on particular platforms because of inherent limitations of the platform. The result is the deployment of incompatible applications, often platform dependent, to meet specific needs in various parts of the enterprise.

Java eliminates this impediment through the deployment of consistent virtual machine platforms -- which can run any Java-compliant application program -- across the enterprise. The entire concept of "porting" an application to different client platforms becomes a thing of the past.

Development Too Slow

With up to 80% of MIS time and resources spent maintaining the current complex set of fat-client applications, not much is left over for quickly developing new applications that drive the business.

Java Computing reduces the time spent in application development, testing and roll-out, leaving more time and money for creating new added-value applications. Instant roll-out of applications on an intranet Web site allows much shorter, more iterative application development cycles, the ability to react quickly to bugs, and much happier users. Java builder tools from a large variety of vendors support much higher programmer productivity. Developers are seeing productivity increases of two to five times over traditional languages such as C or C++. The simplicity of Java, its strong type checking, automatic memory management and exception handling all contribute to this increased productivity.

An Enterprise IT Architecture Based on Java Computing 6

Infrastructure

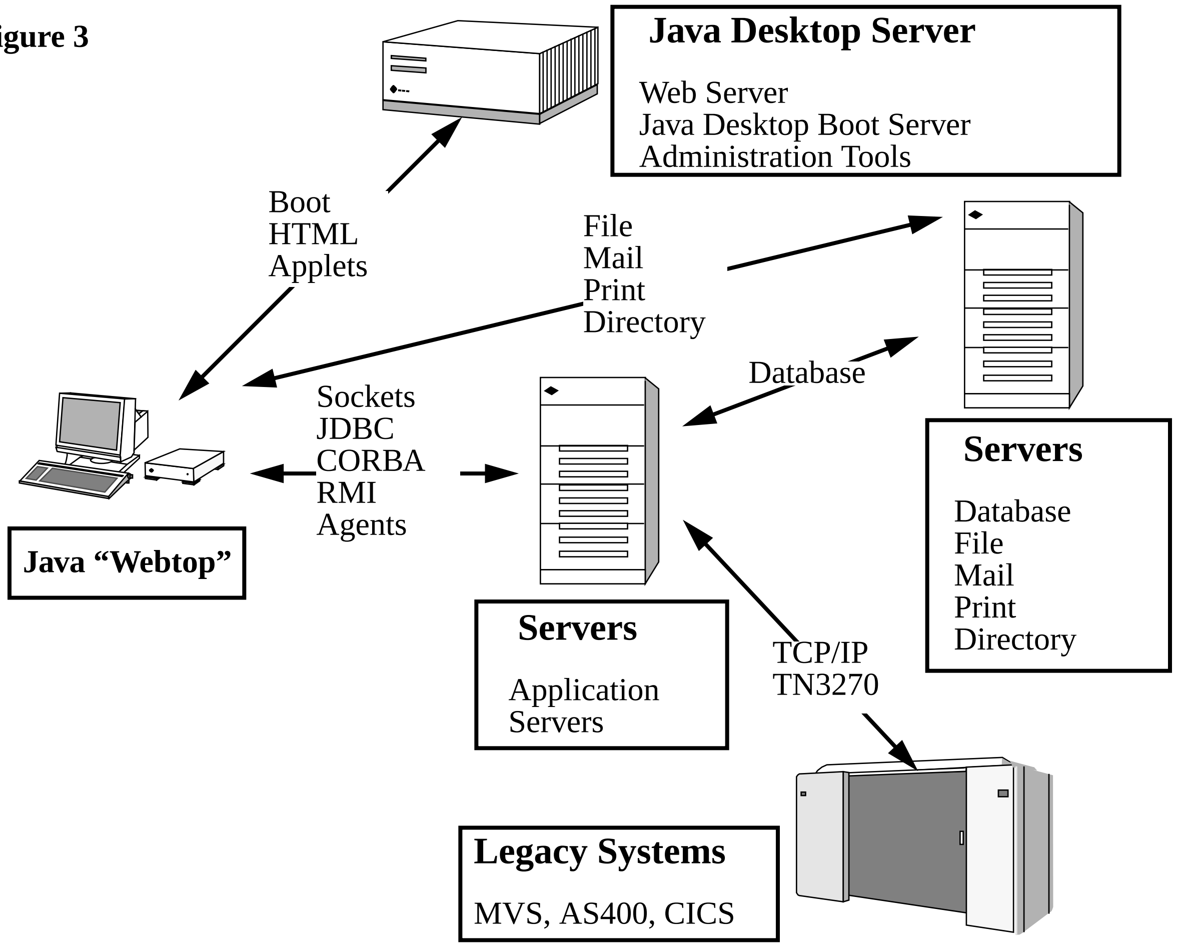
The ultimate Java-based IT architecture will be made up of the following elements:

- *An enterprise network infrastructure, or "intranet."* This consists of a TCP/IP-based network with both LAN (10-Base-T minimum speed), plus WAN connectivity where needed. An intranet may be as simple as a single TCP/IP Ethernet LAN with a single Web server, or it may be a global TCP/IP network with multiple subnets and a high-speed WAN backbone. It may also provide Internet connectivity through a firewall. Open Internet standards have produced a large number of vendors for the intranet market that sell switches, routers, hubs and other network equipment necessary to set up and maintain TCP/IP networks.
- *A Web server, which serves HTML pages and Java applets to the clients in a corporate intranet.* This server may be replicated to increase the number of supported clients without increasing the administrative burden. It is shown as a "Java Desktop Server" in Figure 3, and combines the booting service for thin-client desktops.
- *Distributed application servers, upon which reside business applications or objects to be accessed by clients across the intranet.* The servers may represent a range of price/performance points and be tailored to specific applications, but they share the common features of interfacing with legacy code and databases and delivering application services to Java clients. "Application server" is a logical distinction and may, in fact, be combined with the Web or applet servers mentioned above.
- *Database, file, mail, print and directory services and other dedicated function servers accessible by Java client applets.* Again, in smaller installations, these services may reside on the same server as the Web or applet server.

- *An integrated management system for managing servers, networks and clients, including both fat clients and thin clients.* A good example for a large enterprise is Solstice Enterprise Manager™ from Sun (see http://www.sun.com/solstice/em-products/network/ent.man.html).
- *Legacy systems integrated on the network.* Access to these systems is provided through application server "glue code," or, in the case of legacy applications that do not have Java front ends yet, through Java terminal emulator applets.
- *Many clients (up to 90% in some enterprises based on Sun customer studies) that are dataless (and some diskless) Java devices with access to server applets across the corporate intranet and the Internet, to the extent allowed by network security policies.* These clients would access and cache Java applets dynamically as needed and would store no applications or data locally. Clients that need to access a large number of applications quickly may have disk caches to improve response time, but this local disk would store no permanent data and would require no administration.
- *Some fat client seats.* Their primary applications are very diverse and require large amounts of local data manipulation and storage (e.g., CAE systems).
- *Server management; client processing.* With the Java architecture, local client processing is used to reduce peak network and server loads, while all administration and management tasks are handled centrally on the server.

The total annual cost per Java device seat of this configuration would be in the range of $2,500.

Figure 3



These elements can be deployed very flexibly using a variety of network topologies. For example, a branch office or a customer site may have a local Netra™ Web server for booting and serving Java applets to the local Java devices via a LAN. This branch server would actually be a mirror site of the master Web server at the main office, and all administration for the branch could be done at the main office via the Java front-end Solstice tools. (For Sun's Netra line of Web server products, see "http://www.sun.com/productssolutions/hw/servers/netra.html").

Alternatively, a Java device that contained the Java Virtual Machine in flash ROM could be connected directly to the main office via a dial-up connection.

Development

In building "custom software," there are two primary concerns to address. First, are the best development tool(s) for the job being employed? Second, are the best tool(s) for integrating the new application into the existing IT architecture being used? Since Java has become so widely adopted by software tools vendors, many of the best traditional development and middleware tools are already available for Java development, together with some new tools from new vendors. (A good resource for developers is the "Java Developer Connection" - see http://java.sun.com). Therefore, Java is a good overall software platform choice for building a company's next-generation software.

It is crucial when developing Java applets to make sure that they run on multiple browsers and/or Java device platforms. This ensures that there is no inadvertent dependency on any non-standard APIs. The rich set of Java APIs allows the development of full-featured applets that are truly "write once/run anywhere."

Selecting the best Java development tool generally depends on three criteria: developer experience, application type and developer platform. For example, professional developers with C/C++ experience who need to build a high-performance and complex OLTP application should look at integrated Java development environments such as Sun's Java WorkShop™ (see "http://www.sun.com/developer-products/index.html"). Meanwhile, Web developers with HTML experience who need to build animated Web pages should evaluate Java Web development tools such as DimensionX Liquid Motion. Commercial designers and developers who lack programming experience but want to build database applications should evaluate Java rapid application development tools such as Spider Technologies Net-Dynamics or WebSeQueL from Infospace, Inc.

Java can be used to access existing database and transaction systems as well as for new business applications. To ensure this, Sun extended Java to connect to databases (through the JDBC Java database API), to CORBA-based object servers (through a Java/IDL API) and to other object servers (through a Java Beans API, which integrates Java, ActiveX, OpenDoc and Live Connect objects into a new cross-platform framework). In addition, many of Sun's Java partners have developed other useful connectivity tools, such as Open Connect's Web Connect 3270 emulator written in Java. So, whether one needs to integrate Java applications with databases, legacy environments or object application code, there are many solutions to choose from to create a completely integrated environment.

Deployment

One way of classifying “users” is by the range of computing applications they use. For example, accounts receivable collectors or airline reservation agents may spend their whole day using a small group of related applications. Conversely, an administrative office worker may use a number of applications such as e-mail, word processing, spreadsheets, database inquiries, and Web browsers.

The client “applications” can be segmented by the intensity of the computing resources required for them to function. Certain applications such as complex 3-D design applications (mechanical or electrical CAD) or image manipulation (Adobe Photoshop) are very rich in the quantity of data they manage and/or the computing resources they consume in accessing, manipulating and/or displaying information. Conversely, text-based administration functions and even sophisticated spreadsheets and word processors require more modest computing and I/O performance by today's standards.

The “field” of enterprise computing users can be visualized along the “intensity” and “diversity” dimensions summarized below. Below, we have suggested what we believe is a typical distribution of users across these fields (enterprises will vary in their distribution of users). This data is based on feedback from Sun customers.

“High Intensity/Low Diversity Computing”

- Engineering Technicians, Manufacturing Control Technicians, Dedicated Technical Authors
- Estimated % of Enterprise Users: 10%

“High Intensity/High Diversity Computing”

- Technical Design Engineers, Financial Securities Traders, Computer Animation Artists
- Estimated % of Enterprise Users: 10%

“Low Intensity/Low Diversity Computing”

- Airline Reservations Agents, Accounts Receivable Collectors, Inventory Analysts, Customer Service Representatives
- Estimated % of Enterprise Users: 50%

“Low Intensity/High Diversity Computing”

- Office Professionals, Administrative Assistants, Financial Analysts
- Estimated % of Enterprise Users: 30%

We suggest that all enterprise clients with the exception of the “high intensity/high diversity” group can improve productivity at lower costs through Java. In other words, up to 90% of enterprise clients could be migrated to become a Java client and realize the full potential benefit of Java. This migration can be done gradually and seamlessly, beginning with the use of a Java-enabled browser on existing platforms and later migrating to dedicated Java devices.

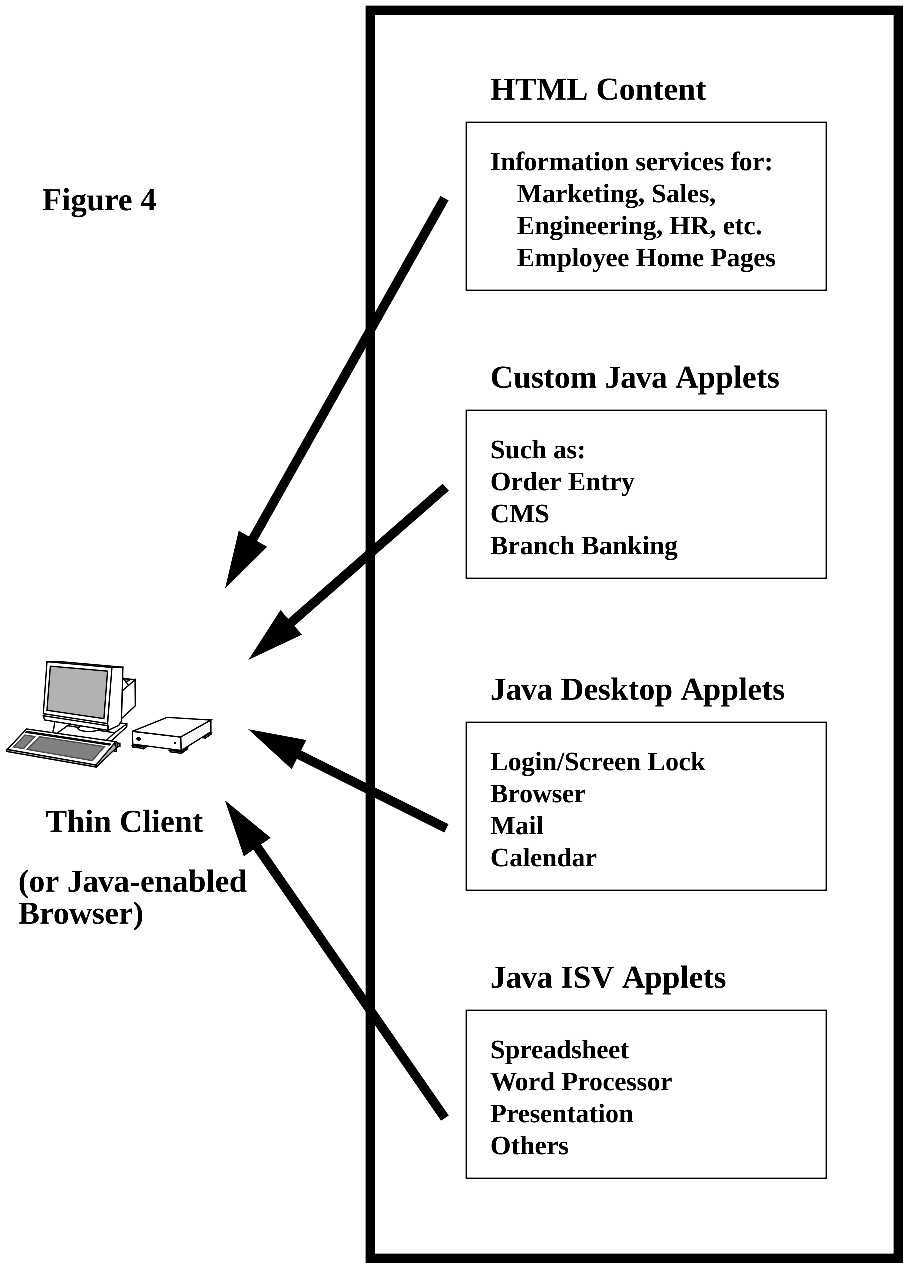
The first step on the way to Java Computing is to make sure every desktop has access to a Java-enabled browser. This platform can initially be used to share information within the company in the form of HTML pages. Since all desktops are Java-enabled, the Java-enabled browser can also be used to roll out corporate applications. An important first application to roll out for the webtop is electronic mail. Moving to Internet standards for mail (IMAP4 and SMTP) simplifies the management of enterprise mail and familiarizes the user community with accessing functionality from the browser. Sun's Solstice Internet Mail provides an IMAP4 and SMTP back end for the enterprise and integrates with existing e-mail systems as well.

The reliability and scalability of the Java Computing environment is dependent on the reliability and scalability of the network and servers. Server hardware and software should support scalable multiprocessing, clustering and RAS features to ensure high availability and performance. For more information, see the Sun RAS Solutions for Mission-Critical Computing white paper at: http://www.sun.com/products-n-solutions/hw/servers/business.html.

Initial deployment of Java devices should target low intensity/low diversity application areas such as order entry systems, call centers, branch banking and reservation systems (Figure 4), using custom applet design to support these applications. These desktops do not require access to productivity applications and are ideal candidates for thin clients.

Finally, when more general productivity applications become available in Java, a wider deployment of Java devices can take place. This need not be forced -- in fact, outmoded fat clients may essentially become thin client equivalents, running only a browser as their environment until they are replaced by less-expensive, easier-to-maintain thin clients. There are a number of efforts underway in the industry to provide full productivity desktop functionality in the Java environment. Java's “write once/run anywhere” advantage is just as appealing to productivity software vendors as it is to corporate MIS departments.

Figure 4


HTML Content
Information services for:
Marketing, Sales,
Engineering, HR, etc.
Employee Home Pages
Custom Java Applets
Such as:
Order Entry
CMS
Branch Banking
Java Desktop Applets
Login/Screen Lock
Browser
Mail
Calendar
Java ISV Applets
Spreadsheet
Word Processor
Presentation
Others
Thin Client
(or Java-enabled Browser)

Management

The key management feature of a Java networking environment is that the manpower needed to manage desktop clients is greatly reduced. All management is accomplished at the server. Since there is no permanent client data or code, there is no client state to manage. A client hardware failure is corrected by a simple swap of desktop machines. Policies specifying which applications can be run and which resources can be accessed are set centrally.

Powerful tools like Solstice Enterprise Manager can manage the servers and clients in this environment. Internet standards like DHCP (Dynamic Host Configuration Protocol) allow automatic IP address assignment to clients and minimize desktop administrative tasks. Additional tools including Solstice Backup/Archive and Solstice DiskSuite™ meet the data backup, reliability and performance needs of the enterprise network.

With desktop Netra Java servers and Solstice network management tools, the system or network administrator can manage all systems from a central location, carrying out tasks such as the addition/removal of users, software configuration control, monitoring and trouble-shooting. These functions are themselves performed through a Java interface, so they can be accessed from any client platform (with proper authorization).

A big advantage of the Java networking model is that there is no need to move desktop computer files when a person is moved. In fact, users may log on anywhere they can access the intranet and have full access to their complete home environment. All permanent state information is maintained in the intranet Web, not on the desktop. This saves a tremendous amount of desktop administration. It also makes possible connected mobile computing with full access to a user's complete environment through authorized dial-up access at home or on the road.

Migration

It will take most enterprises several years to fully migrate to this type of architecture, but the process can be done very easily in small steps without immediately discarding a single computing platform or application. Browsers can run Java applets on legacy clients such as Windows 3.1 or Win95 until the time comes to replace the fat client with a Java device. Terminal applications (e.g., 3270 or VT100, etc.) can be run via Java emulator applets anywhere a Java Virtual Machine is present.

Since thin clients can display any HTML page, forms-based HTML applications can be run from any of these desktops, and HTML can be used as a way to publish information inside the enterprise.

Back-end systems can be left as is and moved from the mainframe to high-end servers over time. Middleware can be written in any language, using off-the-shelf tools from a variety of vendors producing TP monitors, CORBA ORBs, messaging and queuing systems. Since Java Virtual Machines and just-in-time compilers are being included in almost every major operating systems by the vendors, middleware can also be written in Java and gain the benefits of a highly productive language with “write once/run anywhere” capability.

What Can the General Manager and CIO Do Today? 7

While the industry is just beginning the transition to Java, it is clear that significant functional improvements and economic savings will be realized by adopters. The exact level and nature of these savings will become more apparent over time. It is clear, however, that the savings and functional benefits are significant.

At the same time, the general manager and CIO must run their day-to-day business and solve immediate problems. Summarized below are several recommendations for the general manager and CIO that attempt to balance the management of near-term crises against the requirement for long-term structural change in Java-based information management:

Today:

- Build an enterprise intranet infrastructure using Internet technologies.
- Deploy Java client functionality (webtops) throughout the enterprise, initially through browsers such as Netscape Navigator.
- Move electronic mail to the intranet with access through the webtop. This provides universal access to e-mail from all clients using Internet protocols (IMAP4/SMTP), which work over low or high bandwidth. It also introduces end users to general application access through their webtop (Java-enabled browser or thin client).
- Begin a systematic training program in the MIS department to build Java competence. Many programs are already available from Sun and others.
- Set up organizations whose charter is to build Java applications. A number of commercial integrators can provide help in the transition to Java Computing. Examples include Andersen Consulting, Cambridge Technology Partners, EDS and SunIntegration Services.

- Postpone mass upgrading of client operating systems until an evaluation can be made of superior economics and functionality available through Java. Selective upgrades may be necessary for particular applications or users.
- Track product announcements closely as the numerous Java applications under development make their way to market. Many Java applications are already on the market and by the end of 1996, there should be Java applications available in all major application areas.
- Track industry announcements related to the launch of Java device hardware by many vendors in 1996. Selectively purchase and evaluate Java device hardware in a range of application and user areas.
- Question the acquisition of new fat-client hardware (in all areas except high intensity/high diversity computing) until an evaluation of thin client products is completed.

Over next 6 months:

- Identify dedicated-use applications -- desktops where employees run a small set of custom applications all day long as their job.
- Write or re-write these applications in Java and begin using Java devices in these areas.
- Proposals for system or application changes or upgrades should include the evaluation of the Java option.

Over next 12 months:

- Target early migration of applications to Java in areas where the most immediate benefits appear realizable. Continue thin-client deployment on dedicated-function desktops. Areas of potential immediate return include:
    - Dedicated-function desktops (CMS, reservations, order entry, etc.)
    - Customized decision support applications
    - Customer or supplier communications/transactions
    - Employee communications/transactions

  For these last three categories, the ability to access Java applets from any client is a major advantage.

Over next 18 months:

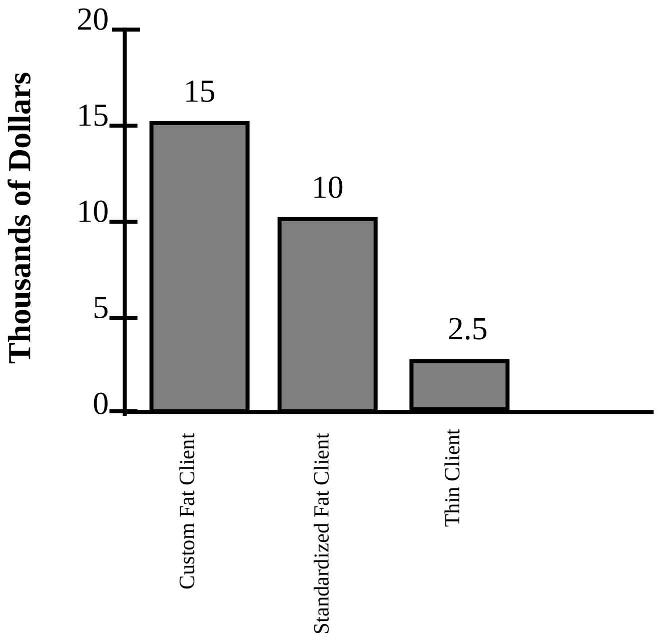
- When a critical mass of Java-based applications has been introduced, begin to replace fat clients with Java devices.

The Cost of Java Computing vs. Fat-Client Computing

A quick expense analysis of Java Computing based upon the points discussed earlier can be conducted for a range of Java device deployment percentages. The chart in Figure 5 shows the typical yearly cost per seat of some of the various kinds of desktops discussed in Section 5.

Desktop Cost/Seat/Year

Figure 5



The most common enterprise desktop environments today are either “custom fat client” (variations in the configuration on each desktop) or “standard fat client” (a common desktop configuration defined and enforced by corporate policy). The yearly cost savings of converting some fraction of these desktops to Java devices is analyzed below.

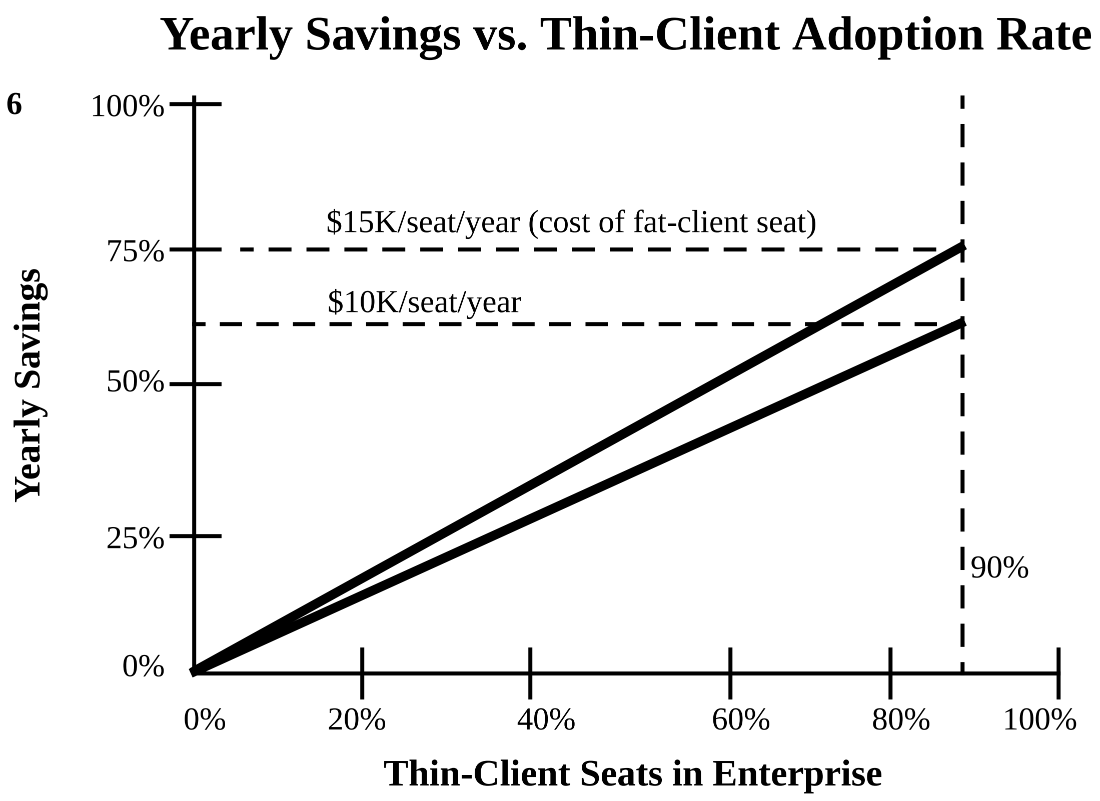
One additional assumption is necessary for this analysis. The one-time cost of a major upgrade of a fat client (e.g., from Windows 3.1 to WindowsNT) has been studied by many CIOs, who have found a wide range of costs associated with the upgrade. An expense level that seems to be supportable by most is in the range of $5,000 per seat. This includes the cost of the software, associated hardware upgrades required, dedicated support costs and end user time consumed in the upgrade process. At the end of this upgrade process, the CIO has added little new capability to the company and has spent a large portion of the MIS budget to implement such an upgrade.

We suggest that, instead of investing in the next upgrade to another fat-client operating system, the enterprise invest that same money into Java migration and quickly begin to capture the savings that can be gained through the Java enterprise computing architecture.

Figure 6 shows the percentage yearly cost savings that can be achieved by converting desktops to Java devices. Two scenarios are shown -- one in which the fat client cost per seat is $15,000/year and one in which the fat client cost per seat is $10,000/year. The first case would correspond to the “custom fat client” enterprise environment noted above, and the second case corresponds to the “standardized fat client.” The percentage savings are shown plotted against the percentage of seats that can be converted to thin-client desktops.

In the optimal case -- 90% of all desktops converted to thin client -- the saving is 75% in an enterprise with custom fat client seats currently deployed and 67.5% in an enterprise with a standardized fat-client environment. Individual enterprises will have different conversion rates, which may start small but grow over time. Even at relatively low rates, the savings are evident.

Figure 6



The primary drivers behind Java Computing savings include:

- Reduced acquisition cost of desktop hardware/software systems
- Reduced complexity at the desktop
  - All OS and application software stored only on servers
  - All software upgrades done at the servers
  - System administration all done at the servers
  - All data files reside on servers
  - Less on-site support infrastructure required
  - Less user self-maintenance required

- Desktop failures can be corrected immediately with a system swap
- Lower costs associated with supporting fewer platforms
- Lower application development cost
- Lower application support infrastructure cost
- Reduced complexity / cost of network security

The amount of the potential savings through Java is staggering. Sun calculated a possible cost savings for converting 75% of existing fat-client desk-tops to thin clients in a typical organization (a larger Fortune 1000 company, the current main market for Java Computing), with 100,000 nodes within the organization that are possible candidates[1] for conversion. The following calculation was based on an estimated total cost per seat per year of $10,000[2] for "standard fat clients" and $2,500 per seat per year for thin clients. (See section 6-1 for discussion of thin-client costs.) *Such an enterprise could save $562.5 million annually deploying Java Computing.*

Using an even more conservative conversion metric of 25% -- which is significantly below the expectations Sun has been given by customers interested in Java Computing -- the savings is $187.5 million. Even this very low conversion rate could produce projected savings that would be seen as a major victory for the CIO in any enterprise. These cost estimates are based on an extrapolation from Sun's internal experience with our 35,000-node TCP/IP network. Cost figures will necessarily vary based on specific circumstances in a given corporation, but we believe the above analysis to be fairly representative. Therefore, we recommend that any medium or large enterprise undertake its own evaluation of Java Computing before investing heavily in upgrading existing PC/Windows fat-client architectures.

[1] Some customers that Sun consulted for feedback on expected conversion rates reported expectations of ultimately converting up to 90% of all their desktops to thin clients. The calculations above use more conservative 75% and 25% conversion rates which would in reality be spread over time.

[2] Using $10,000 / seat / year cost for a fat client, the cost savings per seat is $7,500. This is multiplied by 75,000 seats for a 75% conversion rate, and 25,000 seats for a 25% conversion rate.