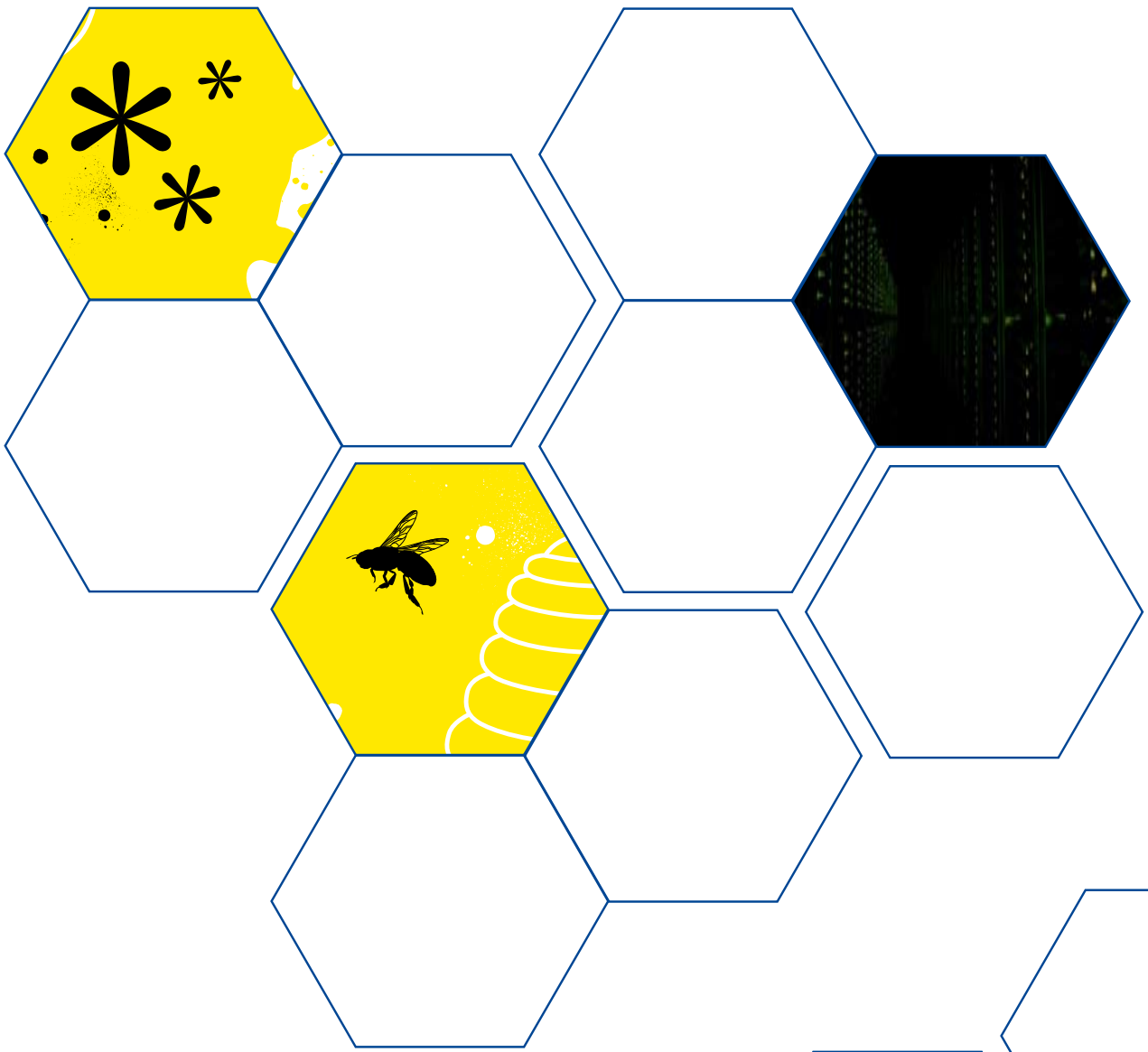
EXECUTIVE SUMMARY

The way the company manages information technology often goes unnoticed. But Google's unconventional approach to information management is both highly effective and highly efficient—and may be the way other organizations deploy technology in the future. **Here's why**.



CASE

216

How Google Works

BY DAVID F. CARR

With his unruly hair dipping across his forehead, Douglas Merrill walks up to the lectern set up in a ballroom of the Arizona Biltmore Resort and Spa, looking like a slightly rumpled university professor about to start a lecture. In fact, he is here on this April morning to talk about his work as director of internal technology for Google to a crowd of chief information officers gathered at a breakfast sponsored by local recruiting firm Phoenix Staffing.

Google, the secretive, extraordinarily successful $6.1 billion global search engine company, is one of the most recognized brands in the world. Yet it selectively discusses its innovative information management infrastructure—which is based on one of the largest distributed computing/grid systems in the world.

Merrill is about to give his audience a rare glimpse into the future according to Google, and explain the workings of the company and the computer systems behind it.

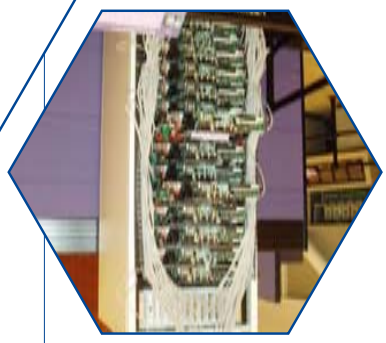
For all the razzle-dazzle surrounding Google—everything from the press it gets for its bring-your-dog-to-work casual workplace, to its stock price, market share, dizzying array of beta product launches and its death-match competition with Microsoft—it must also solve more basic issues like billing, collection, reporting revenue, tracking projects, hiring contractors, recruiting and evaluating employees, and managing video-conferencing systems—in other words, common business problems.

But this does not mean that Google solves these problems in a conventional way, as Merrill is about to explain.

"We're about not ever accepting that the way something has been done in the past is necessarily the best way to do it today," he says.

Among other things, that means that Google often doesn't deploy standard business applications on standard hardware. Instead, it may use the same text parsing technology that drives its search engine to extract application input from an e-mail, rather than a conventional user interface based on data entry forms. Instead of deploying an application to a conventional server, Merrill may deploy it to a proprietary server-clustering infrastructure that runs across its worldwide data centers.

Google runs on hundreds of thousands of servers—by one estimate, in excess of



450,000—racked up in thousands of clusters in dozens of data centers around the world. It has data centers in Dublin, Ireland; in Virginia; and in California, where it just acquired the million-square-foot headquarters it had been leasing. It recently opened a new center in Atlanta, and is currently building two football-field-sized centers in The Dalles, Ore.

By having its servers and data centers distributed geographically, Google delivers faster performance to its worldwide audience, because the speed of the connection between any two computers on the Internet is partly a factor of the speed of light, as well as delays caused by network switches and routers. And although search is still Google's big moneymaker, those servers are also running a fast-expanding family of other applications like Gmail, Blogger, and now even Web-based word processors and spreadsheets.

That's why there is so much speculation about Google the Microsoft-killer, the latest firm nominated to drive everything to the Web and make the Windows desktop irrelevant. Whether or not you believe that, it's certainly true that Google and Microsoft are banging heads. Microsoft expects to make about a $1.5 billion capital investment in server and data structure infrastructure this year. Google is likely to spend at least as much to maintain its lead, following a $838 million investment in 2005.

And at Google, large-scale systems technology is all-important. In 2005, it indexed 8 billion Web pages. Meanwhile, its market share continues to soar. According to a recent ComScore Networks qSearch survey, Google's market share for search among U.S. Internet users reached 43% in April, compared with 28% for Yahoo and 12.9% for The Microsoft Network (MSN). And Google's market share is growing; a year ago, it was 36.5%. The same survey indicates that Americans conducted 6.6 billion searches online in April, up 4% from the previous month. Google sites led the pack with 2.9 billion search queries performed, followed by Yahoo sites (1.9 billion) and MSN-Microsoft (858 million).

This growth is driven by an abundance of scalable technology. As Google noted in its most recent annual report filing with the SEC: "Our business relies on our software and hardware infrastructure, which provides substantial computing resources at low cost. We currently use a combination of off-the-shelf and custom software running on clusters of commodity computers. Our considerable investment in developing this infrastructure has produced several key benefits. It simplifies the storage and processing of large amounts of data, eases the deployment and operation of large-scale global products and services, and automates much of the administration of large-scale clusters of computers."

Google buys, rather than leases, computer equipment for maximum control over its infrastructure. Google chief executive officer Eric Schmidt defended that strategy in a May 31 call with financial analysts. "We believe we get tremendous competitive advantage by essentially building our own infrastructures," he said.

Google does more than simply buy lots of PC-class servers and stuff them in racks, Schmidt said: "We're really building what we think of internally as supercomputers."

Because Google operates at such an extreme scale, it's a system worth studying, particularly if your organization is pursuing or evaluating the grid computing strategy, in which

high-end computing tasks are performed by many low-cost computers working in tandem.

Despite boasting about this infrastructure, Google turned down requests for interviews with its designers, as well as for a follow-up interview with Merrill. Merrill did answer questions during his presentation in Phoenix, however, and the division of the company that sells the Google Search Appliance helped fill in a few blanks.

In general, Google has a split personality when it comes to questions about its back-end systems. To the media, its answer is, "Sorry, we don't talk about our infrastructure." Yet, Google engineers crack the door open wider when addressing computer science audiences, such as rooms full of graduate students whom it is interested in recruiting. As a result, sources for this story included technical presentations available from the University of Washington Web site, as well as other technical conference presentations, and papers published by Google's research arm, Google Labs.

What Other CIOs Can Learn

While few organizations work on the scale of Google, CIOs can learn much from how the company deploys its technology and the clues it provides to the future of technology. Even if Google falters as a company, or competitors and imitators eventually take away its lead, the systems architecture it exemplifies will live on. "Google is a harbinger in the same way that in the early 1960s, a fully loaded IBM 1460 mainframe gave us a taste of more powerful computers to come," says *Baseline* columnist Paul A. Strassmann.

Strassmann, whose career includes senior information-technology management roles at government agencies including the Department of Defense, has been advising the military that it needs to become more like Google. This recommendation is partly inspired by his admiration for the company's skill at rapidly deploying computing power throughout

the world as it's done the past few years. "Clearly, they have the largest [computer] network anywhere," he says.

Among other things, Google has developed the capability to rapidly deploy prefabricated data centers anywhere in the world by packing them into standard 20- or 40-foot shipping containers, Strassmann says. That's just the sort of capability an army on the move would like to have at its disposal to provide tactical support for battle or relief operations.

According to Strassmann, the idea of portable data centers has been kicking around the military for years, but today exists mostly as a collection of PowerPoint slides. And Google-like rapid access to information might be just what's needed in a war zone, where reports from the field too often pile up unread, he says. The military equivalent of Googling your competition would be for a field officer to spend a few seconds typing a query into a mobile device to get the latest intelligence about a hostile village before the troops move in. "It's Google skinned down into the hands of a Marine," he says.

Google's creation of the shipping container data centers was also reported in November by *Triumph of the Nerds* author Robert X. Cringely in his blog at Pbs.org. He describes it as a system of "5,000 Opteron processors and 3.5 petabytes of disk storage that can be dropped off overnight by a tractor-trailer rig. [A petabyte is a quadrillion bytes, the next order of magnitude up from a terabyte, which is a trillion]. The idea is to plant one of these puppies anywhere Google owns access to fiber, basically turning the entire Internet into a giant processing and storage grid." Google will not comment on these reports.

Greg Linden, one of the architects of Amazon.com's early systems, is fascinated by what Google has created but wary of the hype it sometimes attracts. "Google definitely has been influential, changing how many companies think about the computers that power their systems," says Linden, now CEO at Findory.com, a personalized news site he founded. "But it might be going a bit far to say they are leading a revolution by themselves."

Although Google's requirements might seem so exotic that few other organizations would need anything like its technology stack, Linden says he spoke with a hedge fund technology manager who said how much he would love to employ the distrib-

GOOGLE BASE CASE

Headquarters: 1600 Amphitheatre Pkwy., Mountain View, CA 94043

Phone: (650) 253-0000

Business: Web search and supporting information products

Chief Information Officer: Douglas Merrill, vice president, engineering, and senior director of information systems

2005 Financials: $6 billion revenue; $1.46 billion net profit.

Challenge: Deliver technologies to organize the world's information and make it universally accessible and useful.

Baseline Goals:

- Build the systems and infrastructure to support a global, $100 billion company.
- Expand data center infrastructure, which, as measured by the property and equipment, more than doubled, from $379 million in 2004 to $962 million last year.
- Maintain high productivity as measured by revenue per employee, which ranged from $1.38 million to $1.55 million in 2005.

PLAYER ROSTER

Larry Page (above, right)
Co-Founder & President, Products
Sergey Brin (above, left)
Co-Founder & President, Technology

Page devised Google's original search relevancy ranking algorithm, PageRank, while working toward a Ph.D. in computer science from Stanford. He was also heavily involved in the design of Google's early distributed computer design and server racks. Brin's research interests include search engines, information extraction from unstructured sources, and data mining of large text collections and scientific data. The two met while Ph.D. candidates at Stanford, where they collaborated on the Web indexing project that became Google.

Douglas Merrill
Vice President, Engineering, and Senior Director of Information Systems
Merrill plays the role of chief information officer at Google, managing both traditional enterprise systems and the applications developed by Google engineers for internal use. A student of social and political organization who holds a Ph.D. in psychology from Princeton, Merrill worked on computer simulations of education, team dynamics and organizational effectiveness for the Rand Corp. and went on to serve as a senior vice president at Charles Schwab, where he was responsible for, among other things, information security and the company's common infrastructure.

Eric Schmidt
Chairman and CEO
Schmidt shares responsibility for Google's day-to-day management with Page and Brin. Schmidt, who was previously the chief executive officer of Novell and, before that, chief technology officer of Sun Microsystems, was brought into the company in 2001 to give Google a more seasoned technology leader.

Dave Girouard
Vice President and General Manager, Google Enterprise
Girouard leads the division responsible for adapting Google technologies for the enterprise market, with the Google Search Appliance and Google Mini utilities as its primary products. Girouard comes from multimedia search company Virage and previously was a systems consultant for Booz Allen Hamilton and Accenture (then known as Andersen Consulting).

W.M. Coughran Jr.
Vice President of Engineering for Systems Infrastructure
Coughran is responsible for large-scale distributed computing programs underlying Google's products. Since joining Google in early 2003, he has refined the teams that handle Web crawling, storage and other systems.

Urs Hölzle
Senior Vice President, Operations, and Google Fellow
Hölzle was Google's first engineering vice president and led the development of the company's operational infrastructure. He now focuses on keeping it running and improving its performance.

Vinton G. Cerf
Vice President and Chief Internet Evangelist
A "father of the Internet," he is responsible for identifying new enabling technologies and applications for Google.

Terry Winograd
Professor and director of the Human-Computer Interaction Group, Stanford University
An early adviser to the Google co-founders, Winograd joined them in authoring an academic paper on PageRank and worked for the company in 2002 and 2003, while on leave from Stanford. As a specialist in human interaction with computers, he contributed to Google's user interface design in areas such as the display of search results.

BASE TECHNOLOGIES

Google primarily relies on **its own internally developed software** for data and network management and has a reputation for being skeptical of "not invented here" technologies, so relatively few vendors can claim it as a customer.

APPLICATION	PRODUCT	SUPPLIER
Distributed file system	Google File System	Google proprietary
Distributed scheduling	Global Work Queue	Google proprietary
Very large database management systems	BigTable, Berkeley DB	Google proprietary, Sleepycat Software/Oracle
Server operating system	Red Hat Linux (with kernel-level modifications by Google)	Red Hat, Google
Web protocol accelerator	NetScaler Application Delivery	Citrix Systems
Web content translation	Rosette Language Analyzers for Chinese, Japanese and Korean (used in combination with Google proprietary translation technology)	Basis Technology
File conversion and content extraction	Outside In	Stellent

GOOGLE'S PRIMARY PROGRAMMING LANGUAGES INCLUDE C/C++, JAVA AND PYTHON. GUIDO VAN ROSSUM, PYTHON'S CREATOR, WENT TO WORK FOR GOOGLE AT THE END OF 2005. THE COMPANY ALSO HAS CREATED SAWZALL, A SPECIAL-PURPOSE DISTRIBUTED COMPUTING JOB PREPARATION LANGUAGE.

uted computing software behind Google's data centers to run data-intensive market simulations. And Linden says BigTable, Google's system for managing very large databases, sounds like something "I would have loved to have had at Amazon."

It is, however, something that Merrill can tap into at will. One of five engineering vice presidents, Merrill is Google's senior director of information systems—effectively, the CIO.

For some basic corporate functions like financial management, Merrill chooses the same kind of technologies you would find in any other corporate data center. On the other hand, Google doesn't hesitate to create applications for internal use and put them on its own server grid. If the company's software engineers think they can tinker up something to make themselves more productive—for example, a custom-built project tracking system—Merrill doesn't stand in their way.

The Start of the Story

Google started with a research project into the structure of the Web led by two Stanford University Ph.D. candidates, Larry Page and Sergey Brin. After initially offering to sell the search engine they had created to an established firm, such as Yahoo, but failing to find a buyer, they established Google in 1998 to commercialize the technology. For the first few years of the company's existence, the co-founders were determined to avoid making money through advertising, which they thought would corrupt the integrity of search results. But when their initial technology licensing business model fell flat, they compromised. While keeping the Google home page ad-free, they began inserting text ads next to their search results, with ad selection driven by the search keywords. The ads were clearly separated from the search results, but highly effective because of their relevance. At pennies per click, the ad revenue pouring into Google began mounting quickly.

When Google went public in 2004, analysts, competitors and investors were stunned to learn how much money the company with the ad-free home page was raking in—$1.4 billion in the first half of 2004 alone. Last year, revenue topped $6 billion.

Google and its information-technology infrastructure had humble beginnings, as Merrill illustrates early in his talk with a slide-show photo of Google.stanford.edu, the academic research project version of the search engine from about 1997, before the formation of Google Inc., when the server infrastructure consisted of a jumble of PCs scavenged from around campus.

"Would any of you be really proud to have this in your data center?" Merrill asks, pointing to the disorderly stack of servers connected by a tangle of cables. "But this is the start of the story," he adds, part of an approach that says "don't necessarily do it the way everyone else did. Just find some way of doing it cheap and effectively—so we can learn."

The basic tasks that Google had to perform in 1997 are the same it must perform today. First, it scours the Web for content, "crawling" from one page to another, following links. Copies of pages it finds must be stored in a document repository, along with their Internet addresses, and further analyzed to produce indexes of words and links occurring in each document. When someone types keywords into Google, the search engine compares them against the index to determine the best matches and displays links to them, along with relevant excerpts from the cached Web documents. To make all this work, Google had to store and analyze a sizable fraction of all the content on the Web, which posed both technical and economic challenges.

By 1999, the Google.com search engine was running in professionally managed Internet data centers run by companies like Exodus. But the equipment Google was parking there was, if anything, more unconventional, based on hand-built racks with corkboard trays. The hardware design was led by Page, a natural engineer who once built a working ink-jet printer out of Legos. His team assembled racks of bare motherboards, mounted four to a shelf on corkboard, with cheap no-name hard drives purchased at Fry's Electronics. These were packed close together (like "blade servers before there were blade servers," Merrill says). The individual servers on these racks exchanged and replicated data over a snarl of Ethernet cables plugged into Hewlett-Packard network switches. The first Google.com production system ran on about 30 of these racks. You can see one for yourself at the Computer History Museum, just a few blocks away from Google's Mountain View headquarters.

Part of the inspiration behind this tightly packed configuration was that, at the time, data centers were charging by the square foot, and Page wanted to fit the maximum computer power into the smallest amount of space. Frills like server enclosures around the circuit boards would have just gotten in the way.

This picture makes the data center manager in Merrill shudder. "Like, the cable management is really terrible," he

says. Why aren't cables carefully color-coded and tied off? Why is the server numbering scheme so incoherent?

The computer components going into those racks were also purchased for rock-bottom cost, rather than reliability. Hard drives were of a "poorer quality than you would put into your kid's computer at home," Merrill says, along with possibly defective memory boards sold at a fire-sale discount. But that was just part of the strategy of getting a lot of computing power without spending a lot of money. Page, Brin and the other Google developers started with the assumption that components would fail regularly and designed their search engine software to work around that.

Google's co-founders knew the Web was growing rapidly, so they would have to work very hard and very smart to make sure their index of Web pages could keep up. They pinned their hopes on falling prices for processors, memory chips and storage, which they believed would continue to fall even faster than the Web was growing. Back at the very beginning, they were trying to build a search engine on scavenged university resources. Even a little later, when they were launching their company on the strength of a $100,000 investment from Sun Microsystems co-founder Andy Bechtolsheim, they had to make their money stretch as far as possible. So, they built their system from the cheapest parts money would buy.

At the time, many dot-com companies, flush with cash, were buying high-end Sun servers with features like RAID hard drives. RAID, for redundant arrays of independent disks, boosts the reliability of a storage device through internal redundancy and automatic error correction. Google decided to do the same thing by different means—it would make entire computers redundant with each other, making many frugally constructed computers work in parallel to deliver high performance at a low cost.

By 1999, when Google received $25 million in venture funding, the frugality that had driven the early systems design wasn't as much of a necessity. But by then, it had become a habit.

Later Google data centers tidied up the cabling, and corkboard (which turned out to pose a fire hazard) vanished from the server racks. Google also discovered that there were downsides to its approach, such as the intense cooling and power requirements of densely packed servers.

In a 2003 paper, Google noted that power requirements of a densely packed server rack could range from 400 to 700 watts per square foot, yet most commercial data centers could support no more than 150 watts per square foot. In response, Google was investigating more power-efficient hardware, and reportedly switched from Intel to AMD processors for this reason. Google has not confirmed the choice of AMD, which was reported earlier this year by Morgan Stanley analyst Mark Edelstone.

Although Google's infrastructure has gone through many changes since the company's launch, the basic clustered server rack design continues to this day. "With each iteration, we got better at doing this, and doing it our way," Merrill says.

A Role Model

For Google, the point of having lots of servers is to get them working in parallel. For any given search, thousands of computers work simultaneously to deliver an answer.

For CIOs to understand why parallel processing makes so much sense for search, consider how long it would take for one person to find all the occurrences of the phrase "service-oriented architecture" in the latest issue of *Baseline*. Now assign the same task to 100 people, giving each of them one page to scan, and have them report their results back to a team leader who will compile the results. You ought to get your answer close to 100 times faster.

On the other hand, if one of the workers on this project turns out to be dyslexic, or suddenly drops dead, completion of the overall search job could be delayed or return bad results. So, when designing a parallel computing system—particularly one like Google's, built from commodity components rather than top-shelf hardware—it's critical to build in error correction and fault tolerance (the ability of a system to recover from failures).

Some basic concepts Google would use to scale up had already been defined by 1998, when Page and Brin published their paper on "The Anatomy of a Large-Scale Hypertextual Web Search Engine." In its incarnation as a Stanford research project, Google had already collected more than 25 million Web pages. Each document was scanned to produce one index of the words it contained and their frequency, and that, in turn, was transformed into an "inverted index" relating keywords to the pages in which they occurred.

But Google did more than just analyze occurrences of keywords. Page had come up with a formula he dubbed PageRank to improve the relevance of search results by analyzing the link structure of the Web. His idea was that, like frequent citations of an academic paper, links to a Web site were clues to its importance. If a site with a lot of incoming links—like the Yahoo home page—linked to a particular page, that link would also carry more weight. Google would eventually have to modify and augment Page's original formula to battle search engine spam tactics. Still, PageRank helped establish Google's reputation for delivering better search results.

Previous search engines had not analyzed links in such a systematic way. According to *The Google Story,* a book by *Washington Post* writer David Vise and Mark Malseed, Page had noticed that early search engine king AltaVista listed the number of links associated with a page in its search results but didn't seem to be making any other use of them. Page saw untapped potential. In

addition to recording which pages linked to which other pages, he designed Google to analyze the anchor text—the text of a link on a Web page, typically displayed to a Web site visitor with underlining and color—as an additional clue to the content of the target page. That way, it could sometimes divine that a particular page was relevant even though the search keywords didn't appear on that page. For example, a search for "cardiopulmonary resuscitation" might find a page that exclusively uses the abbreviation "CPR" because one of the Web pages linking to it spells out the term in the link's anchor text.

All this analysis requires a lot of storage. Even back at Stanford, the Web document repository alone was up to 148 gigabytes, reduced to 54 gigabytes through file compression, and the total storage required, including the indexes and link database, was about 109 gigabytes. That may not sound like much today, when you can buy a Dell laptop with a 120-gigabyte hard drive, but in the late 1990s commodity PC hard drives maxed out at about 10 gigabytes.

To cope with these demands, Page and Brin developed a virtual file system that treated the hard drives on multiple computers as one big pool of storage. They called it BigFiles. Rather than save a file to a particular computer, they would save it to BigFiles, which in turn would locate an available chunk of disk space on one of the computers in the server cluster and give the file to that computer to store, while keeping track of which files were stored on which computer. This was the start of what essentially became a distributed computing software infrastructure that runs on top of Linux.

The overall design of Google's software infrastructure reflects the principles of grid computing, with its emphasis on using many cheap computers working in parallel to achieve supercomputer-like results.

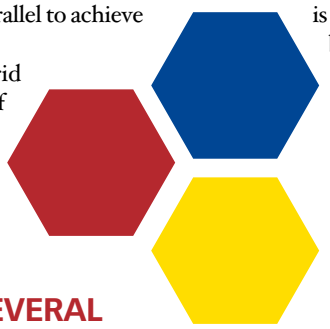
Some definitions of what makes a grid a grid would exclude Google as having too much of a homogenous, centrally controlled infrastructure, compared with a grid that teams up computers running multiple operating systems and owned by different organizations.

But the company follows many of the principles of grid architecture, such as the goal of minimizing network bottlenecks by minimizing data transmission from one computer to another.

Instead, whenever possible, processing instructions are sent to the server or servers containing the relevant data, and only the results are returned over the network.

"They've gotten to the point where they're distributing what really should be considered single computers across continents," says Colin Wheeler, an information systems consultant with experience in corporate grid computing projects, including a project for the Royal Bank of Canada.

Having studied Google's publications, he notes that the company has had to tinker with computer science fundamentals in a way that few enterprises would: "I mean, who writes their own file system these days?"



FILES CAN BE SEVERAL GIGABYTES. TO MANAGE DISK SPACE EFFICIENTLY, THE GFS ORGANIZES DATA INTO 64-MEGABYTE CHUNKS.

The Google File System

In 2003, Google's research arm, Google Labs, published a paper on the Google File System (GFS), which appears to be a successor to the BigFiles system Page and Brin wrote about back at Stanford, as revamped by the systems engineers they hired after forming Google. The new document covered the requirements of Google's distributed file system in more detail, while also outlining other aspects of the company's systems such as the scheduling of batch processes and recovery from subsystem failures.

The idea is to "store data reliably even in the presence of unreliable machines," says Google Labs distinguished engineer Jeffrey Dean, who discussed the system in a 2004 presentation available by Webcast from the University of Washington.

For example, the GFS ensures that for every file, at least three copies are stored on different computers in a given server cluster. That means if a computer program tries to read a file from one of those computers, and it fails to respond within a few milliseconds, at least two others will be able to fulfill the request. Such redundancy is important because Google's search system regularly experiences "application bugs, operating system bugs, human errors, and the failures of disks, memory, connectors, networking and power supplies," according to the paper.

The files managed by the system typically range from 100 megabytes to several gigabytes. So, to manage disk space efficiently, the GFS organizes data into 64-megabyte "chunks," which are roughly analogous to the "blocks" on a conventional file system—the smallest unit of data the system is designed to support. For comparison, a typical Linux block size is 4,096 bytes. It's the difference between making each block big enough to store a few pages of text, versus several fat shelves full of books.

To store a 128-megabyte file, the GFS would use two chunks. On the other hand, a 1-megabyte file would use one 64-megabyte chunk, leaving most of it empty, because such "small" files are so rare in Google's world that they're not worth worrying about (files more commonly consume multiple 64-megabyte chunks).

A GFS cluster consists of a master server and hundreds or thousands of "chunkservers," the computers that actually store the data. The master server contains all the metadata, including file names, sizes and locations. When an application requests a given file, the master server provides the addresses of the relevant chunkservers. The master also listens for a "heartbeat" from the chunkservers it manages—if the heartbeat stops, the master assigns another server to pick up the slack.

In technical presentations, Google talks about running more than 50 GFS clusters, with thousands of servers per cluster, managing petabytes of data.

More recently, Google has enhanced its software infrastructure with BigTable, a super-sized database management system it developed, which Dean described in an October presentation at the University of Washington. BigTable stores structured data used by applications such as Google Maps, Google Earth and My Search History. Although Google does

use standard relational databases, such as MySQL, the volume and variety of data Google manages drove it to create its own database engine. BigTable database tables are broken into smaller pieces called tablets that can be stored on different computers in a GFS cluster, allowing the system to manage tables that are too big to fit on a single server.

Reducing Complexity

Google's distributed storage architecture for data is combined with distributed execution of the software that parses and analyzes it.

To keep software developers from spending too much time on the arcana of distributed programming, Google invented MapReduce as a way of simplifying the process. According to a 2004 Google Labs paper, without MapReduce the company found "the issues of how to parallelize the computation, distribute the data and handle failures" tended to obscure the simplest computation "with large amounts of complex code."

Much as the GFS offers an interface for storage across multiple servers, MapReduce takes programming instructions and assigns them to be executed in parallel on many computers. It breaks calculations into two parts—a first stage, which produces a set of intermediate results, and a second, which computes a final answer. The concept comes from functional programming languages such as Lisp (Google's version is implemented in C++, with interfaces to Java and Python).

A typical first-week training assignment for a new programmer hired by Google is to write a software routine that uses MapReduce to count all occurrences of words in a set of Web documents. In that case, the "map" would involve tallying all occurrences of each word on each page—not bothering to add them at this stage, just ticking off records for each one like hash marks on a sheet of scratch paper. The programmer would then write a reduce function to do the math—in this case, taking the scratch paper data, the intermediate results, and producing a count for the number of times each word occurs on each page.

One example, from a Google developer presentation, shows how the phrase "to be or not to be" would move through this process.

MAP

key	TO	BE	OR	NOT	TO	BE
value	1	1	1	1	1	1

REDUCE

key	TO	BE	OR	NOT
value	2	2	1	1

While this might seem trivial, it's the kind of calculation Google performs ad infinitum. More important, the general technique can be applied to many statistical analysis problems. In principle, it could be applied to other data mining problems that might exist within your company, such as searching for recurring categories of complaints in warranty claims against your products. But it's particularly key for Google, which invests heavily in a statistical style of computing, not just for search but for solving other problems like automatic translation between human languages such as English and Arabic (using common patterns drawn from existing translations of words and phrases to divine the rules for producing new translations).

MapReduce includes its own middleware—server software that automatically breaks computing jobs apart and puts them back together. This is similar to the way a Java programmer relies on the Java Virtual Machine to handle memory management, in contrast with languages like C++ that make the programmer responsible for manually allocating and releasing computer memory. In the case of MapReduce, the programmer is freed from defining how a computation will be divided among the servers in a Google cluster.

Typically, programs incorporating MapReduce load large quantities of data, which are then broken up into pieces of 16 to 64 megabytes. The MapReduce run-time system creates duplicate copies of each map or reduce function, picks idle worker machines to perform them and tracks the results.

Worker machines load their assigned piece of input data, process it into a structure of key-value pairs, and notify the master when the mapped data is ready to be sorted and passed to a reduce function. In this way, the map and reduce functions alternate chewing through the data until all of it has been processed. An answer is then returned to the client application.

If something goes wrong along the way, and a worker fails to return the results of its map or reduce calculation, the master reassigns it to another computer.

As of October, Google was running about 3,000 computing jobs per day through MapReduce, representing thousands of machine-days, according to a presentation by Dean. Among other things, these batch routines analyze the latest Web pages and update Google's indexes.

Google's Secrets

For all the papers it has published, Google refuses to answer many questions. "We generally don't talk about our strategy ... because it's strategic," Page told *Time* magazine when interviewed for a Feb. 20 cover story.

One of the technologies Google has made public, PageRank, is Page's approach to ranking pages based on the interlinked structure of the Web. It has become one of the most famous elements of Google's technology because he published a paper on it, including the mathematical formula. Stanford holds the patent, but through 2011 Google has an exclusive license to PageRank.

Still, Yahoo's research arm was able to treat PageRank as fair game for a couple of its own papers about how PageRank might be improved upon; for example, with its own TrustRank variation based on the idea that trustworthy sites tend to link to other trustworthy sites. Even if competitors can't use PageRank per se, the information Page published while still at Stanford gave competitors a starting point to create something similar.

"PageRank is well known because Larry published it—well, they'll never do that again," observes Simson Garfinkel, a postdoctoral fellow at Harvard's Center for Research on Computation and Society, and an authority on information

HOW GOOGLE MANAGES ITS GLOBAL WORKFORCE

GOOGLE'S DOUGLAS MERRILL, a corporate technology director with a background in psychology, is firmly convinced that while technology plays a role in keeping projects on track, so do the Ping-Pong tables and the elaborate cafeteria at company headquarters in Mountain View, Calif.—the social settings that get coders away from their desks and talking to each other about their projects.

As he explained during a presentation in Phoenix, that's one of the things that made Google successful as a startup, but now the challenge is to maintain that quirky culture as the company grows. It now has more than 6,800 employees and dozens of sales and engineering offices around the world.

Having overseas engineering offices is proving to be important because people who are based France, or India, or China tend to do a better job of localizing Google's applications—that is, not just translating the user interface but changing the way the system works to meet local requirements.

E-mail is a pretty good answer for tracking projects, Merrill says, but e-mail alone is not sufficient for really managing them, and not for keeping the people who are working on them happy and productive. "At the end of the day, face time rules," he says.

The closest thing to a technological answer to reaching those overseas workers is videoconferencing, and even then face-to-face contact is still better. So, Google has had to hire a cadre of managers who spend most of their time on planes, in addition to hiring local managers for those offices. That goes against the grain for a company that prides itself on its flat management structure, but it's necessary, Merrill says.

Remote offices also tend to suffer disproportionately from minor inconsistencies in technology infrastructure, which can hobble efforts at collaboration between engineers in different offices. For example, an application might depend on a certain configuration of Lightweight Directory Access Protocol (LDAP) servers, which are used to store information about networks, devices and user accounts. "We don't really want to spend days tracking down problems in code because the LDAP schema that we use in Dublin is different than the LDAP schema that we use in New York," Merrill says. "Those sorts of things used to happen. The time we spent doing that cost money, it cost productivity and, most importantly, it cost frustration."

Trying to dictate the proper configuration from the home office, and have remote offices set things up to those specifications, didn't work, he says. Instead, he developed an "office-in-a-box," with LDAP servers and other basic infrastructure preconfigured at headquarters and shipped into the field. (This almost sounds like the prefab data center in a container Google is reported to have created, except that this is more the size of a steamer trunk.)

"We took the data center, we shrunk it, put it in a brightly colored box and rolled one into every office," Merrill explains. "Then we built a whole bunch of tools that work in the background to verify that the LDAP change made in one place is replicated to all the other offices-in-a-box around the world."

Keeping the technology synchronized is one way of keeping people and projects in sync, but there are always limits to what it can accomplish, he says: "That's Merrill's Law: There are no technological solutions to social problems." —D.F.C.

security and Internet privacy. Today, Google seems to have created a very effective "cult of secrecy," he says. "People I know go to Google, and I never hear from them again."

Because Google, which now employs more than 6,800, is hiring so many talented computer scientists from academia—according to *The Mercury News* in San Jose, it hires on average 12 new employees a day and recently listed 1,800 open jobs—it must offer them some freedom to publish, Garfinkel says. He has studied the GFS paper and finds it "really interesting because of what it doesn't say and what it glosses over. At one point, they say it's important to have each file replicated on more than three computers, but they don't say how many more. At the time, maybe the data was on 50 computers. Or maybe it was three computers in each cluster." And although the GFS may be one important part of the architecture, "there are probably seven layers [of undisclosed technology] between the GFS system and what users are seeing."

One of Google's biggest secrets is exactly how many servers it has deployed. Officially, Google says the last confirmed statistic for the number of servers it operates was 10,000. In his 2005 book *The Google Legacy,* Infonortics analyst Stephen E. Arnold puts the consensus number at 150,000 to 170,000. He also says Google recently shifted from using about a dozen data centers with 10,000 or more servers to some 60 data centers, each with fewer machines. A *New York Times* report from June put the best guess at 450,000 servers for Google, as opposed to 200,000 for Microsoft.

The exact number of servers in Google's arsenal is "irrelevant," Garfinkel says. "Anybody can buy a lot of servers. The real point is that they have developed software and management techniques for managing large numbers of commodity systems, as opposed to the fundamentally different route Microsoft and Yahoo went."

Other Web operations like Yahoo that launched earlier built their infrastructure around smaller numbers of relatively high-end servers, according to Garfinkel. In addition to saving money, Google's approach is better because "machines fail, and they fail whether you buy expensive machines or cheap machines."

Of particular interest to CIOs is one widely cited estimate that Google enjoys a 3-to-1 price-performance advantage over its competitors—that is, that its competitors spend $3 for every $1 Google spends to deliver a comparable amount of computing power. This comes from a paper Google engineers published in 2003, comparing the cost of an eight-processor server with that of a rack of 176 two-processor servers that delivers 22 times more processor power and three times as much memory for a third of the cost. In this example, the eight-processor server was supposed to represent the traditional approach to delivering high performance, compared with Google's relatively cheap servers, which at the time used twin Intel Xeon processors.

But although Google executives often claim to enjoy a price-performance advantage over their competitors, the company doesn't necessarily claim that it's a 3-to-1 difference. The numbers in the 2003 paper were based on a hypothetical comparison, not actual benchmarks versus competitors, according to Google. Microsoft and Yahoo have also had a few years to react with their own cost-cutting moves.

GOOGLE COURTS THE ENTERPRISE

WHILE THERE IS NO DOUBT ABOUT THE POWER GOOGLE.COM commands among advertisers and Webmasters, Google the enterprise vendor is another thing entirely.

Since 99% of Google's $6 billion in revenue continues to come from advertising, the Google Enterprise division represents a tiny part of the overall business. This is the group that wants to sell you a little bit of Google in a box—the Search Appliance product line—that embeds a variation of the Google.com search engine software that powers Google.com in a yellow server box or blue blade server that enterprise customers can plug into their own data centers.

THE APPLIANCE PRODUCT LINE INCLUDES THE ENTRY-LEVEL GOOGLE MINI, which starts at $1,995 for a model capable of indexing 100,000 documents. It is often used to provide the search capability for public Web sites, although it is also used internally by small enterprises or departments of larger ones. Beyond that level, the Search Appliance line includes the Google Appliance GB-1001, which can handle up to a million documents; and the GB-5005 and GB-8008, which, when delivered in the form of multiple servers in a rack and working together as a Google File System cluster, can handle many millions of documents. "It really is like a little Google data center in a box," says Matt Glotzbach, head of products for Google Enterprise.

Because the technology is delivered in the form of an appliance, customers aren't supposed to crack open the case and tinker with the technology inside, and they aren't provided with root access to the server operating environment.

Instead, Google provides a set of Web-based administration screens, as well as application programming interfaces for modifying the appliance's behavior. The most significant development on that front is **THE INTRODUCTION OF THE ONEBOX APPLICATION PROGRAMMING INTERFACE**, which allows data drawn from other systems that otherwise wouldn't be indexed by the search appliance to be displayed at the top of the search results.

The enterprise version of OneBox is modeled after the feature of Google's public Web site that inserts links to data from weather reports, phone listings or maps into search results when the search engine recognizes a pattern associated with an address, a city name or a phone number in the keywords entered by the user. Similarly, the appliance can be programmed to recognize patterns associated with purchase order numbers or common business queries, and insert links to related data. For example, a search for quarterly sales data could go beyond searching intranet Web content and pop up a link to a more structured data source, such as a Cognos financial analysis application.

Dave Girouard, vice president and general manager of the Google Enterprise business unit, says Google has been beefing up its capabilities to address enterprise requirements, making the appliances easier to install, use and manage.

HOWEVER, GOOGLE STILL NEEDS TO DO A BETTER JOB OF ADDRESSING ENTERPRISE REQUIREMENTS, particularly in terms of support, according to Gartner's Whit Andrews, an authority on the evolution of search technology. "It has taken Google a while to recognize that it needs to do business with the enterprise in a different way from how it does business with the advertiser," he says.

Enterprises that have bought the appliances often give positive reports on the value Google delivers for the money and on the ease of setup and administration, Andrews says. But support is another story, according to Andrews, who has talked to customers who say Google's response to problems with the appliances is too often along the lines of, "Yeah, we know about that, we'll get back to you."

Google says it is investing in improved support. Andrews agrees that the support is better, but says it's still not enough.

However, Google can point to many happy customers. Brown Rudnick Berlack Israels, an international law firm based in Boston, got the Google Mini it purchased up and running in about an hour, says Keith Schultz, who manages the firm's Web sites. "We've really had no problems with it at all," Schultz says.—D.F.C.

"Google is very proud of the cost of its infrastructure, but we've also driven to a very low cost," says Lars Rabbe, who as chief information officer at Yahoo is responsible for all data center operations.

Microsoft provided a written statement disputing the idea that Google enjoys a large price-performance advantage. Microsoft uses clusters of inexpensive computers where it makes sense, but it also uses high-end multi-processor systems where that provides an advantage, according to the statement. And Windows does a fine job of supporting both, according to Microsoft.

Certainly, Yahoo's systems design is different from Google's, Rabbe says. "Google grew up doing one thing only" and wound up with a search-driven architecture "that is very uniform, with a lot of parallelism." Yahoo has increased its focus on the use of parallel computing with smaller servers, he says, but is likely to continue to have a more heterogeneous server infrastructure. For example, Yahoo launched the company on the FreeBSD version of Unix but has mixed in Linux and Windows servers to support specific applications. While the Yahoo and Google companies compete as search engines, he says, "We also do 150 other things, and some of those require a very different type of environment."

Still, Gartner analyst Ray Valdes believes Google retains an advantage in price-performance, as well as in overall computing power. "I buy that. Even though I'm usually pretty cynical and skeptical, as far as I know, nobody has gone to that extent and pushed the envelope in the way they have," he says. "Google is still doing stuff that others are not doing."

The advantage will erode over time, and Google will eventually run up against the limits of how much homegrown technology it can manage, Valdes predicts: "The maintenance of their own system software will become more of a drag to them."

But Google doesn't buy this traditional argument for why enterprises should stick to application-level code and leave more fundamental technologies like file systems, operating systems and databases to specialized vendors. Google's leaders clearly believe they are running a systems engineering company to compete with the best of them.

Exotic but Not Unique

Google's systems seem to work well for Google. But if you could run your own systems on the Google File System, would you want to? Or is this an architecture only a search engine could love?

Distributed file systems have been around since the 1980s, when the Sun Microsystems Network File System and the Andrew File System, developed at Carnegie Mellon University, first appeared. Software engineer and blogger Jeff Darcy says the system has a lot in common with the HighRoad system he worked on at EMC. However, he notes that Google's decision to "relax" conventional requirements for file system data

consistency in the pursuit of performance makes its system "unsuitable for a wide variety of potential applications." And because it doesn't support operating system integration, it's really more of a storage utility than a file system per se, he says. To a large extent, Google's design strikes him as more of a synthesis of many prior efforts in distributed storage.

Despite those caveats, Darcy says he also sees many aspects of the GFS as "cool and useful," and gives Google credit for "bringing things that might have been done mostly as research projects and turning them into a system stable enough and complete enough to be used in commercial infrastructure."

Google software engineers considered and rejected modifying an existing distributed file system because they felt they had different design priorities, revolving around redundant storage of massive amounts of data on cheap and relatively unreliable computers.

Despite having published details on technologies like the Google File System, Google has not released the software as open source and shows little interest in selling it. The only way it is available to another enterprise is in embedded form—if you buy a high-end version of the Google Search Appliance, one that is delivered as a rack of servers, you get Google's technology for managing that cluster as part of the package.

However, the developers working on Nutch, an Apache Software Foundation open-source search engine, have created a distributed software environment called Hadoop that includes a distributed file system and implementation of MapReduce inspired by Google's work.

Google, the Enterprise

In the filing for its 2004 IPO, Google included a letter from Page and Brin that declared, "Google is not a conventional company. We do not intend to become one." Maybe so, but Google still has to satisfy conventional corporate requirements for managing money and complying with laws.

In 2001, Page and Brin hired a more seasoned technology executive to be Google's official corporate leader—Eric Schmidt, formerly chief technology officer of Sun Microsystems and then CEO of Novell.

As recounted in *The Google Story,* one of the first battles Schmidt had to fight was to bring in Oracle Financials to control the company's finances. Page and Brin had been using Quicken, a financial management system for individuals and small businesses. But by this time, Google had 200 employees and $20 million in revenue.

The book quotes Schmidt as saying, "That was a huge fight. They couldn't imagine why it made sense to pay all that money to Oracle when Quicken was available."

So, Google does employ conventional enterprise technology—sometimes. But if you want to understand how the kind of proprietary technology Google possesses can be employed within an enterprise, look at how it's used within Google.

In public presentations, Merrill tries to put Google's technology in the context of how he is using it to address basic business processes like interviewing and hiring the best people, tracking their performance and coordinating projects. "We say our mission is to organize information and make it universally accessible and useful," Merrill says. "We had to figure out how to apply that internally as well."

Consider how Google handles project management. Every week, every Google technologist receives an automatically generated e-mail message asking, essentially, what did you do this week and what do you plan to do next week? This homegrown project management system parses the answer it gets back and extracts information to be used for follow-up. So, next week, Merrill explains, the system will ask, "Last week, you said you would do these six things. Did you get them done?"

A more traditional project tracking application would use a form to make users plug the data into different fields and checkboxes, giving the computer more structured data to process. But instead of making things easier for the computer, Google's approach is to make things easier for the user and make the computer work harder. Employees submit their reports as an unstructured e-mail, and the project tracking software works to "understand" the content of those e-mail notes in the same way that Google's search engine extracts context and meaning from Web pages.

If Google employees found the project tracking system to be a hassle to work with, they probably wouldn't use it, regardless of whether it was supposed to be mandatory, Merrill says. But because it's as easy as reading and responding to an e-mail, "We get pretty high compliance."

Those project tracking reports go into a repository—searchable, of course—so that managers can dip in at any time for an overview on the progress of various efforts. Other Google employees can troll around in there as well and register their interest in a project they want to track, regardless of whether they have any official connection to that project.

"What we're looking for here is lots of accidental cross-pollination," Merrill explains, so that employees in different offices, perhaps in different countries, can find out about other projects that might be relevant to their own work. Despite Google's reputation for secrecy toward outsiders, internally the watchword is "living out loud," Merrill says. "Everything we do is a 360-degree public discussion."

The company takes a more traditional approach with recording financial transactions, however. "Hey, I want those revenues, I really do," he says. That means running financial management software on servers with more conventional virtues like "disks that don't fail very often," he says.

On the other hand, Google runs internally developed human-resources systems on the clustered server architecture, and that's been working fine, according to Merrill: "In general, because of the price-performance trade-off, under current market conditions I can get about a 1,000-fold computer power increase at about 33 times lower cost if I go to the failure-prone infrastructure. So, if I can do that, I will."

Numbers like those ought to catch the attention of any technology manager. Of course, it's true that most corporate enterprises don't run the same applications at the same scale that Google does, and few would find it worthwhile to invest in tinkering with file systems and other systems engineering fundamentals.

But as much as it has poured effort into perfecting its use of those technologies, Google did not invent distributed computing. Companies willing to experiment with commercially available and open-source products for grid computing, distributed file systems and distributed processing may be able to find their own route to Google-like results.

Dossier

OPEN-SOURCE SOFTWARE

Red Hat: **Still Savvy**

Forging ahead with the same business model for more than 12 years might seem old hat to some in the constantly changing world of information technology, but business customers say Red Hat wears it well.

The Raleigh, N.C.-based firm has been selling and supporting open-source Linux software since its founding in 1993. Aside from Google, Red Hat's customers include Amazon.com, Lehman Brothers and Yahoo, and most buyers continue to tip their fedoras to Red Hat's support and pricing.

But while Red Hat has plenty to brag about—its revenues are up more than 80% annually in the past two years, and the company recently acquired open-source middleware company JBoss—some customers say competitor Novell and its SUSE Linux software is nipping at Red Hat's heels.

But the JBoss acquisition helped strengthen Red Hat's relationship with Fiserv Investment Support Services (Fiserv ISS), a Denver-based financial services firm.

Fiserv ISS needed a platform to run its new data warehouse and turned to Red Hat. Rick Kendall, the firm's chief information officer, says Red Hat delivered on the promise of high-level support and consulting throughout the project. And Red Hat's Enterprise Linux worked with two back-end systems that Fiserv ISS had inherited when the company was created from four smaller firms.

The company had also been using a JBoss application server, but having both the Red Hat and JBoss products under one roof was a plus for Kendall. Today, Fiserv ISS has 35 different servers running Red Hat software, and Kendall says he'll expand the deployment over the next five years.

"This is core blocking-and-tackling technology infrastructure that has to work," Kendall says. "And it has."

The City of Chicago is another happy customer, though platform architect Amy Niersbach says Novell appears to have made improvements, such as running Oracle databases and improving its support, since three years ago, when she opted for Red Hat.

In 2003, the Windy City started a pilot program with the Red Hat Enterprise Linux platform to power the city's Business & Information Services Department, which supports enterprise applications throughout Chicago.

Niersbach and company replaced older Sun Solaris servers and installed Red Hat on DL580 G2 servers from Hewlett-Packard. She says the $65,000 she pays annually for Red Hat licenses and support on 65 servers is one-fourth the price of competitors'

FISERV INVESTMENT SUPPORT SERVICES PLANS TO INCREASE ITS USE OF RED HAT SOFTWARE IN THE NEXT FIVE YEARS FOLLOWING A DATA WAREHOUSING PROJECT, SAYS CIO RICK KENDALL.

products. She also says the Red Hat products have needed little or no maintenance since installation.

Jonathan Minter, director of information-technology development and engineering at Liberty University in Lynchburg, Va., bought into Red Hat but isn't 100% sold.

Minter, charged in 2004 with creating a more manageable environment for the school's bustling Web operations, opted for Red Hat because of its reputation for efficient Web serving and its Global File System, an operating system feature that manages server clusters like the ones in the school's new storage area network.

The tool helped cut the number of hours per week needed to manage the environment to 5, from 15 to 20, since the project was completed in April, according to Minter. He also credits Red Hat's consultants with teaching his team about the software's ins and outs.

Still, despite a positive experience with Red Hat, Minter says he's keeping his eye on Novell SUSE Linux—which has added a server management tool—and other competitors. "We're a Red Hat campus for now," Minter says. "Not saying we'll be a Red Hat campus forever."

—BRIAN P. WATSON

Red Hat: At A Glance

1801 VARSITY DRIVE / RALEIGH, NC 27606 / (919) 754-3700 / WWW.REDHAT.COM

TICKER: RHAT (NASDAQ) **EMPLOYEES:** 1,300

MATTHEW J. SZULIK Chairman, CEO & President
PAUL CORMIER EVP, Engineering
PRODUCTS Enterprise Linux 4 gives businesses an open-source computing platform on a subscription basis, with new versions released every 18 months. Global File System manages and connects clustered servers.

FINANCIALS*

	2006FYTD	2005FY	2004FY
Revenue	$278.33M	$151.13M	$82.41M
Net income	$79.69M	$45.43M	$13.73M
R&D spending	82.6%	80.4%	72.8%

* FISCAL YEAR ENDS FEB. 28.

REFERENCE CHECKS

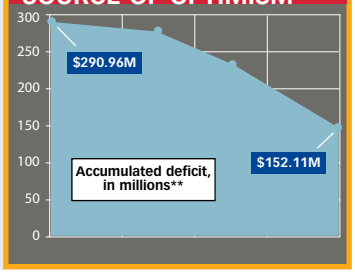
FISERV INVESTMENT SUPPORT SERVICES
Rick Kendall
CIO
Rick.kendall@fiserviss.com

CITY OF CHICAGO
Amy Niersbach
Platform Architect
(312) 744-8735

HERITAGE PROPANE PARTNERS
Mark Wilson
Dir., I.T.
(406) 442-9759 x151

LIBERTY UNIVERSITY
Jonathan Minter
Dir., Engineering and Development
jbminter@liberty.edu

SOURCE OF OPTIMISM



** ACCUMULATED DEFICIT REPRESENTS NET LOSS TO DATE SOURCE: COMPANY REPORTS