Architecture and Design Of High Volume Web Sites

(A Brief History of IBM Sport and Event Web Sites)

Paul Dantzig
IBM T.J. Watson Research Center
19 Skyline Dr.
Hawthorne, NY 10532
001 (914) 723-5686

pauldantzig@us.ibm.com

ABSTRACT

Architecting and designing high volume Web sites has changed immensely over the last six years. These changes include the availability of inexpensive Pentium based servers, Linux, Java applications, commodity switches, connection management and caching engines, bandwidth price reductions, content distribution services, and many others. This paper describes the evolution of the best practices within IBM in architecting sites that handle millions of page views per day. Discussed is the transition to multi-tiered architectures, the use of publish/subscribe software to reduce Web site hits, the migration from static to dynamic content, and techniques for caching of dynamic and personalized content.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software, *Distributed Systems, Performance evaluation (efficiency and effectiveness)*. H.3.5 [**Online Information Services**]: Online Information Services, Web-*Based Services*.

General Terms

Performance, Design, Reliability.

Keywords

Web Serving, Content Management, Content Distribution, Proxy Caching, Java Messaging Services, Publish and Subscribe

1. INTRODUCTION

Architecting and designing high volume Web sites has changed immensely over the last six years. Popular Web sites today can receive a million or more hits per minute. If a significant percentage of those requests require dynamically generated data, those requests can consume orders of magnitude more resources than requests for static data. The initial design of high volume sites consisted of clustering a set of Web servers together and either using round-robin Domain Name Servers [6,8] (RR-DNS) or routing at the TCP level. All the Web servers were fully replicated mirrors of each other (containing the same database, application software, Web serving software, etc.). Users of such systems often experienced inconsistencies in a site's content due to lags or failures in replication as different servers served each page. The cost of replicating databases on the serving nodes consumed cycles needed for serving, resulting in inconsistent response times.

To reduce the consistency and performance issues sites began implementing two tiered architectures. The first tier consisted of a set of systems containing just Web servers and second tier consisted of backend servers such as databases, payment systems, and other legacy systems. This type of architecture provided significant relief from the above problems. Reduced numbers of database replications improved consistency and consumed significantly fewer resources. Response time for static requests was not impacted by the dynamic requests, which were now run in the backend systems. Some high volume sites grew to contain hundreds of front-end nodes. Scaling the backend usually consisted of installing bigger and bigger machines to avoid the database replication problems. It was quickly realized the primary difficulty of running hundreds of nodes was not the hardware cost but the personnel cost to manage them.

Many companies entered the Web server accelerator market (Cisco, Cacheflow, Network Appliance, IBM, etc.) to market new devices to reduce the number of servers. These devices typically handle an order of magnitude or more traffic than a single Web server running on a general-purpose operating system. These devices run as reverse proxy servers with large amounts of memory and disk. In addition, specialized switches and routers came on the market to distribute connections to the Web servers or the Web server accelerators. Newer devices on the market function both as proxies and connection routers, which use heuristics such as number of connections, server load, content, to distribute connections to multiple servers.

In addition to specialized hardware, new software techniques were developed to improve the performance of dynamic pages. One technique that was quite radical at the time was to cache dynamic pages the first time they were created. Methods for constructing pages from individual fragments allowed expensive content to be generated once and shared among many pages. However, successful deployment of these techniques requires much stricter cache consistency. In order to maintain the consistency of the cached data, the server needed to explicitly manage the cache

The Fourteenth International Conference on Software Engineering and Knowledge Engineering ' 00, July 15-19, 2002, Ischia, Italy.

contents. Algorithms for keeping cached dynamic data consistent are described in the referenced papers. [3,4]

Dynamic data was realized to be nothing more than rapidly changing static data. To keep certain types of content up to date required reducing the expiration times to tens of seconds. Some Web sites adopted automatic refresh of pages creating enormous numbers of hits. The new problem was how to reduce the number of hits without sacrificing the better response time users had become accustomed to.

For some types of content, in particular sports scoring results and financial market data, even better response times were considered necessary. Sport and Financial consoles have been created which invoke Java Applets, which open a long-term connection to a "publish and subscribe" system using either private protocols or published protocols such as Java Message Service (JMS) [10]. As content changes the publish and subscribe system pushes changes to those users interested in the change. These systems have become enormously popular with hundred of thousands of active consoles being supported.

Many high volume Web sites require 100% availability. Geographically redundant sites and content distribution services (such as Akamai) place servers and proxy servers closer to the user's ISP connection. In addition to improving availability, using multiple geographic sites or content distribution services can reduce the latency for delivery of graphic and streaming content.

The following sections of this paper describe how the IBM Sport and Event Web sites have been impacted by the above technologies.

2. 1996 Atlanta

Like many first of a kind Web projects during the early years the 1996 Atlanta Games Web Site [11] was architected, designed, and programmed in a hurry. The project started in September of 1995 and was never "completed". Some development continued even as the games were running. Much of the architecture and software came from a Web based digital library joint project between IBM and The Florida Center for Library Automation in Gainesville, Florida. [7] Figure 1 shows the basic architecture for the part of the Atlanta site for handling results on the Web.

The basic system setup was a cluster of twenty IBM RS6000s; rack mounted in two SP2 Frames, and two experimental TCP/IP router boxes for routing the incoming HTTP traffic. Each node contained identical software including the Apache HTTP server, a Fast CGI Apache extension, a connection manager daemon, a cache manager daemon, a database, and twenty pre-started processes preconnected to the database that performed SQL queries in the database if the memory cache did not contain the pre-built page. Each sport and the medals information were allocated a separate cache memory. If a node were restarted the caches would come up completely empty.

The first request for a page would invoke the following actions:

1. The HTTP requests were distributed to a server node based on system load from the TCP/IP Router to the Apache process.
2. Apache called the Fast CGI Apache Extension.
3. The Fast CGI Apache Extension queried the Cache Manager with the unique cache ID (part of the URL).
4. If the page was cached in the appropriate Sport or Medal Cache the cache manager returned the data, which was then returned to the user.
5. If the page was not cached; the connection manger was called to negotiate for a pre-started database process if available.
6. The pre-started process negotiated for and retrieved information from database and generated an HTML result page. The Cache Manager was invoked to store information in the appropriate cache. Finally the page was returned to the user.

In order to maintain cache consistency, whenever an update was made to one of the database tables related to a Sport or Medal a database trigger invoked a process to delete all entries in the specific cache for the affected sport.

As indicated in the introduction, node-to-node consistency and performance problems were both quite visible to the user community. The characteristics of sporting events made the performance problem even more obvious. Users would note when an event would end and inundate the system with requests. This was precisely when the result arrived in the database, causing the entire sport cache to be invalidated. All the requests would have to be resolved from the database until that data finally arrived in the cache. There was more than an order of magnitude difference between a cache hit and a database hit in response times.

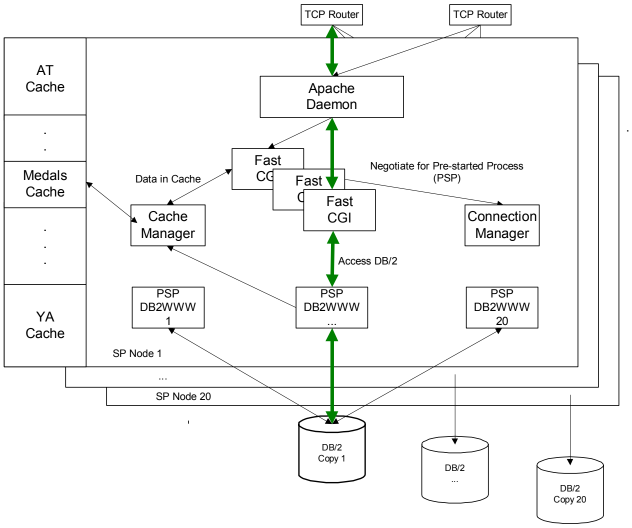


Figure 1. 1996 Atlanta Web Site Architecture

Over the 17 days the 1996 Atlanta site received over 190 million hits. 114 million of those hits were result requests. Peak result server hits was 9 million on August 1, 1996. Cache hit ratio averaged 80% and peak of 45 cache transactions per second per node was achieved.

3. 1998 Nagano

By 1998 IBM had signed up to sponsor several major sporting and other events. These new events included the US Open Tennis Tournament, Wimbledon, the French Open Tennis Tournament, the Grammy Awards, the Tony Awards, etc. It was decided to invest in building a permanent infrastructure that could last several years to handle the Sport and Event Web Sites. This new infrastructure was to be architected, designed, built, and tested before the Nagano games [12].

To achieve a high level of redundancy three permanent geographic locations were chosen close to major Internet backbone sites. One East Coast and two Midwestern United States sites were constructed. In addition a fourth temporary site for Asian traffic was established in Tokyo. Each site consisted of either three or four racks of eleven RS6000 nodes (one SP2 Frame). Figure 2 shows the configuration of one rack of nodes.

One of the criticisms of the Atlanta Web site was that finding information required too many page fetches. The Atlanta site was organized by results, news, photos, etc., and by sports and events. Although country information and athlete biographies were provided, results corresponding to a particular country or athlete could not be collated [2]. Consequently, the Nagano Web site organized results, news, photos, etc. by countries and athletes as well as by sports and events. Figure 3 shows the level of detail contained in each athlete and country page. Each page contained recent results, news, photos, and quotes. The Web results displayed all of the information related to an event (the Atlanta site in many cases only had summaries). Some of the pages now were complex enough to require up to thirty seconds to produce the HTML tables (Ice Hockey results are a notable example). To reduce computational load Country and Athlete pages were only generated every fifteen minutes.

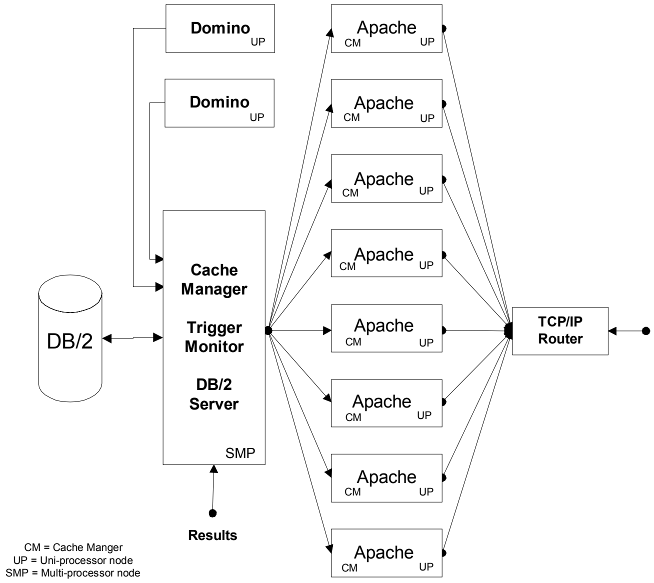
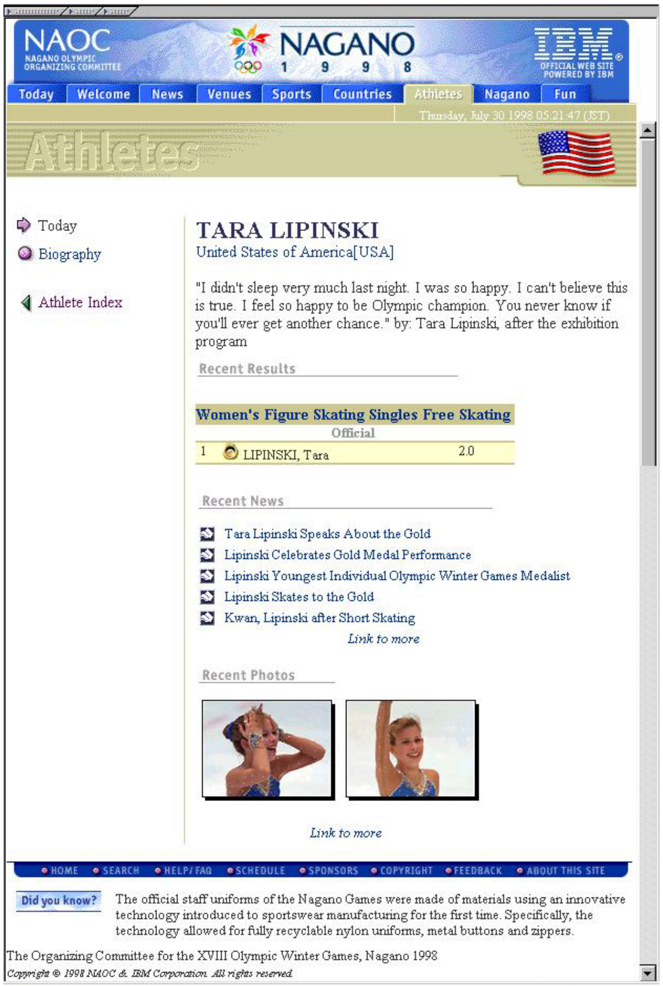


Figure 2. 1998 Nagano Web Site Architecture

Shown in Figure 2 is the Nagano architecture. This architecture is a combination of two architectures: one to pre-generate content and perform content distribution, and the other to serve Web requests. Each time a change was made to news, photos, and related content, the Domino server triggered a change. Each time a schedule, bio, or result changed the database triggered a change.



Thursday, July 30 1998 05:21:47 (JST)

TARA LIPINSKI
United States of America[USA]

"I didn't sleep very much last night. I was so happy. I can't believe this is true. I feel so happy to be Olympic champion. You never know if you'll ever get another chance." by: Tara Lipinski, after the exhibition program

Recent Results

Women's Figure Skating Singles Free Skating		
Official		
1	LIPINSKI, Tara	2.0

Recent News

- Tara Lipinski Speaks About the Gold
- Lipinski Celebrates Gold Medal Performance
- Lipinski Youngest Individual Olympic Winter Games Medalist
- Lipinski Skates to the Gold
- Kwan, Lipinski after Short Skating

Link to more

Recent Photos

Link to more

Did you know? The official staff uniforms of the Nagano Games were made of materials using an innovative technology introduced to sportswear manufacturing for the first time. Specifically, the technology allowed for fully recyclable nylon uniforms, metal buttons and zippers.

The Organizing Committee for the XVIII Olympic Winter Games, Nagano 1998
Copyright © 1998 NAOC & IBM Corporation. All rights reserved.

Figure 3. 1998 Nagano Athlete Page

The pieces of information flowing into the system (each news item, result, etc.), when converted to a Web object (HTML, image, etc.) were now considered a fragment of information and were usable anywhere in the site. A new technology (Object Dependency Graph (ODG)) was developed to keep track of the relationship between fragments and pages. [3,4]

The Cache Manager on each node took on the role of a distributed file system with no persistence. In theory 100% cache hit-ratio could have been achieved with this type of system. Because it was a memory-based cache, pages infrequently requested, such as athlete biographies and country information pages were frequently invalidated. Except for the country and athlete pages, the new model became pro-active page replacement in which a page is replaced in cache and never invalidated. Using fragments and the Object Dependency Graph, only the pages affected by the trigger were updated. By separating page rendering from page serving each server node now had consistent response time. Note: When an SMP node goes down the cache information is lost for all the nodes; to avoid performance problems key pages in the site were primed before the nodes were put back online.

Based on the lessons learned from Atlanta, database replications were minimized. Each site contained no more than four databases (one for each rack). Users were directed to one of four sites by using standard router algorithms (e.g. Japanese traffic to Tokyo, British traffic to the East Coast). If one site failed or needed to be down for maintenance the routers automatically shifted load to the other sites.[2] The only data inconsistencies were between sites so users were unaware of the fact that the Asian site and East Coast site were not consistent with each other.

Nagano was an immense success. All of the goals for the site were achieved: 100% availability, good and consistent response times, every country and athlete treated equally, and no single points of failure. At the end of the event the publishing system contained approximately 50,000 pages in English and Japanese. The site had built 10,645 news article fragments, 14,355 photo fragments, 5,988 result fragments and total combination of the previous into 23,939 Sport, Country, and Athlete pages. The average number of pages rebuilt each day was 20,000 and on the peak day 58,000 pages were rebuilt.

The traffic to the site broke two Guinness records [9]. ***The Most Popular Internet Event Ever Recorded*** based on the officially audited figure of 634.7 million requests over the 16 days of the Games and ***The Most Hits On An Internet Site In One Minute*** based on the officially audited figure of 110,414 hits received in a single minute around the time of the Women's Figure Skating Free Skating (see Figure 3). Hits on the peak day were 55.4 million.

4. 1999 Wimbledon

Even before Nagano, many of the large Web sites were already looking for cheaper and faster ways of serving static content. By 1997 many of them had switched from using expensive proprietary Unix systems (SUN, IBM, HP) to Intel based systems running operating systems like Free-BSD and Linux with Apache to reduce costs and improve performance. In 1997 IBM Research prototyped a combination router and proxy server cache that could support an order of magnitude more hits that a server on the Sport and Event Web sites could handle. The product was transferred to development and made available to the Sport and Event environment after Nagano was complete.

After a number of trial runs during events in 1998 four production Web server accelerators (IBM 2216s) were put into production for the 1999 Wimbledon event [13]. Figure 4 is a graph of the Web hits to the Wimbledon site on July 2, 1999. Along the bottom edge of the graph is time the 10:00 to 16:00 GMT. At about 10:45 the Bethesda Web server accelerator #1 achieved a rate of 66K hits per minute. At 13:32 the four Web server accelerators combined achieved a rate of 140K hits per minute. The Schaumburg, Columbus, and Bethesda bars on the graph are the combined total of 53 Web server nodes. The peek point 13:20 on the graph is 375K hits per minute (i.e. one web server accelerator is equivalent to about fifteen serving nodes).

To get high cache hit ratios (93%) result-scoring content was pushed into the Web server accelerators thru a proprietary interface whenever content changed using the Trigger Monitor daemon. The proxy server built-in to each Web Server Accelerator requested other site content. Each Web server accelerator contained 500M of memory and the working set for Wimbledon was less than 200M of memory.

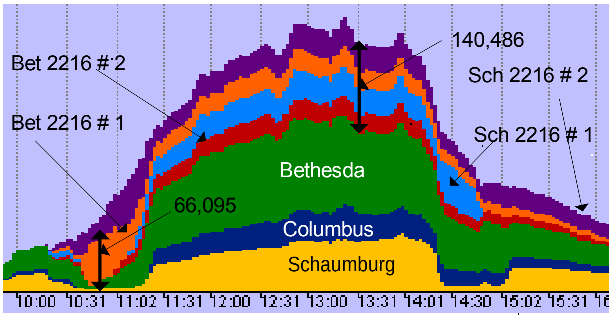


Figure 4. 1999 Wimbledon July 2, 1999 Hits Per Minute

The 1999 Wimbledon event ended with the officially audited figure of 942 million requests over the 14-day event. The peak-hit rate of 430K hits per minute was received on 6/30/99. 71.2 million page views and 8.7 million visits to the site were received. This event proved that inexpensive Web Server Accelerator hardware could reduce the size and cost of the infrastructure even with the enormous rate of growth of these events.

Many Web server accelerators (including the IBM 2216) produce no Web server logs either because the logging function does not exist, or because logging impacts the performance of the accelerator. However, most do support SNMP interfaces, which can be used to record hits, misses, memory usage, and so on. The information for Figure 4 was obtained using the SNMP interface. Without standard Web server logs information such as page views, unique visitors, visits, etc. would be lost. To retain this type of information every page in our event Web sites was modified to add a new request called UC.GIF (uncachable one pel gif). This request is forced to go all the way through to the server. Added to the request are parameters that identify the user, which page was requested, implicit hits, etc. The Web server log processing programs add this information to the standard statistics collected for the site.

5. 2000 Sydney

Based on what we had learned from the previous events the key to the success of Sydney [14] would be making Web Server Accelerator cache misses low cost. Two critical pieces were already prototyped. Learned from Nagano was how to make a system for pre-building pages (Trigger Monitor) and distributing pages. Learned from Wimbledon was how to make a really good edge-serving infrastructure using Web Server Accelerators. Both of these systems needed critical improvements to handle the bigger loads anticipated in Sydney.

Installed after Wimbledon were sixty-five Web server accelerators. A fourth temporary server site in Sydney was planned to augment the three permanent server sites which now had one East Coast, one Midwest, and one West Coast United States locations. An agreement was made with a telecom to collocate a large number of the Web server accelerators at five major Internet hubs in the USA. The installed capacity of the accelerators was between 3 and 4 million hits per minute. Figure 5 shows the server and Web server accelerator locations. The East

Coast would handle the European traffic and the West Coast the Asian traffic. The origin servers were located at IBM Global Services Hosting locations nearby. Experiments had been run during other events, which demonstrated that if **ALL** of the content (HTML, GIFs, JPEGs, etc.) were given **correct expirations,** content would not have to be pushed to the caches; the accelerators would simply proxy the content on a miss. The cache-hit ratio typically dropped to about 75%, which was within the hit capacity of the server sites (Figure 6).

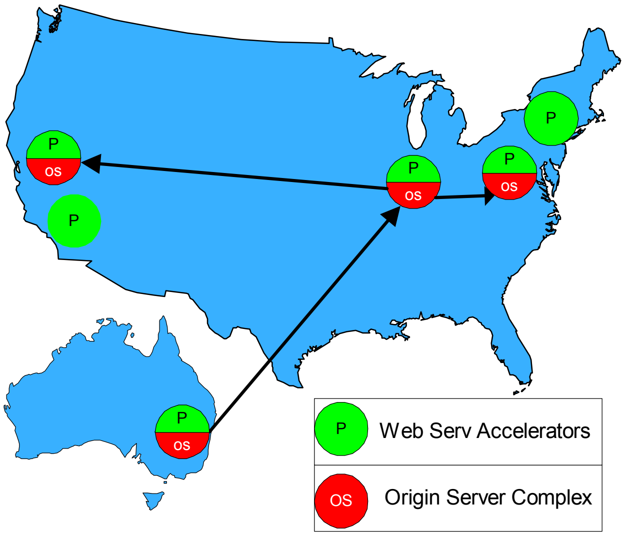


Figure 5. Sydney Origin Servers and Web Server Accelerators Locations

Because of the experimental nature of the dynamic page generation system, both the Atlanta and Nagano sites were designed and implemented as two separate sites, one of which served relatively static content, and the highly dynamic results delivery system. Shortly before the games started an ad-hoc effort was made to merge the sites and present the illusion of a single unified site. It became quite clear during the merge phase of the Nagano site that the Sydney site would need to be constructed from the beginning as a single, unified site.

The combination of building a complete, unified site based on fragment technology, as well as improvements in the technology itself produced an unforeseen problem: how to manage a very large number of fragments. Early estimates for the size of Sydney content were based on simply taking the number of sports, events, countries, athletes, etc. and multiplying Nagano times three (75K pages and 150K rebuilt pages a day). No more than four fragments per page and no imbedded fragments were supported in the Nagano publishing system.

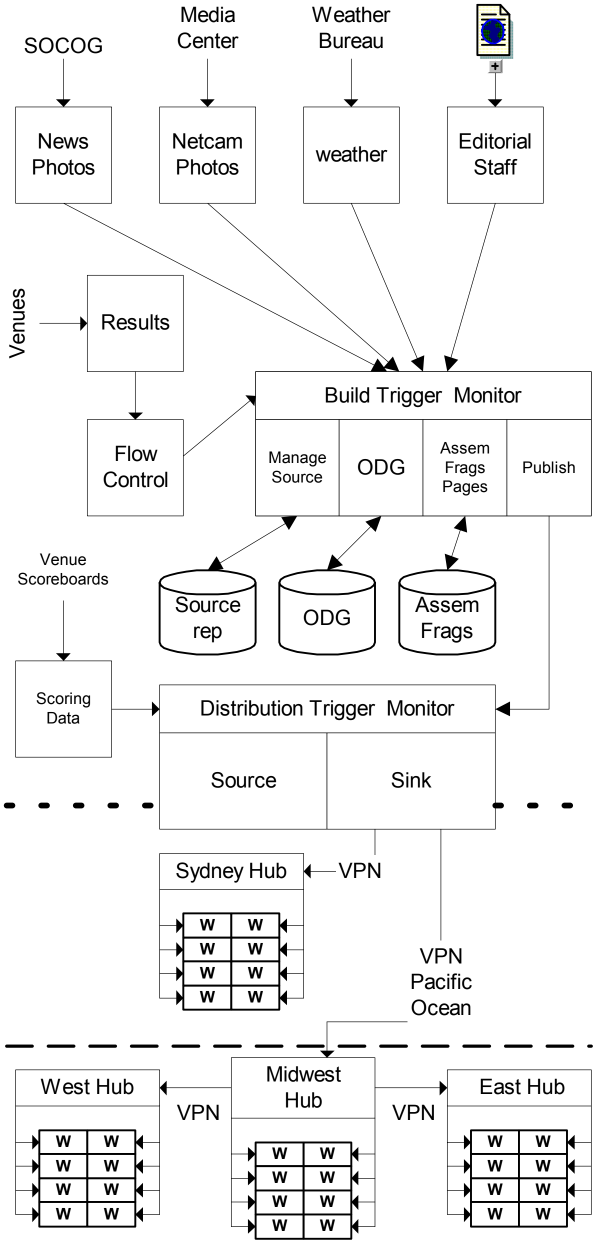


Figure 6. Sydney Content Management and Distribution

Some of the design concept pages for Sydney had thirty fragments on a page with a multiple levels of imbedded fragments. Some updates to the content during the games were expected to affect hundreds of pages (For example: Baseball and Basketball player statistics). To reduce the time required to build these large and complex page updates (Figure 5) source fragments, the ODG, and assembled fragments were saved in a persistent repository called Hash Table on Disk (HTOD). HTOD is a lightweight repository for objects that is both faster and easier to manage than files or a database. [5]

The Trigger Monitor daemon from Nagano was completely rewritten in Java and put into production nine months before the games to insure that all content for the site could be managed by a single content management system.

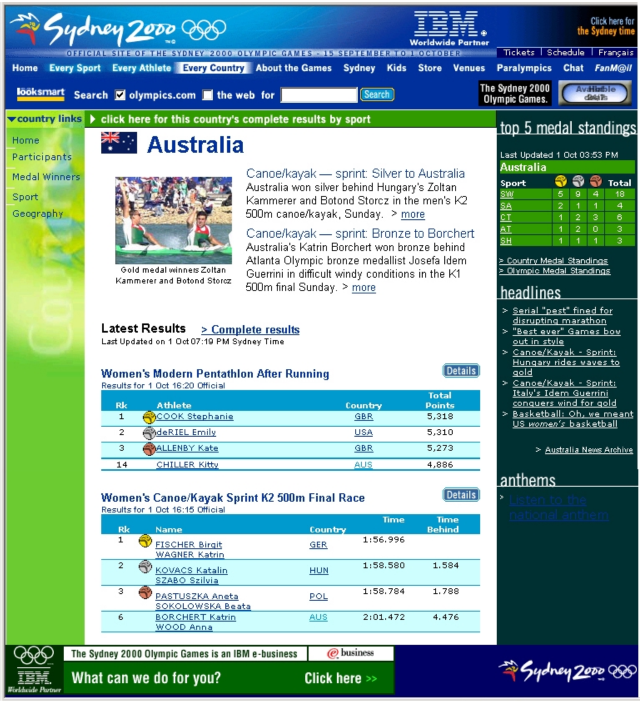
The content distribution system was also redesigned. In Atlanta the results were delivered to each node for database update processing by using a distributed file system. Static content was also served from the distributed file system. In Nagano, the caching system performed the task of the distributed file system for the fast changing results while Web objects such as news, photos, and quotes, which changed at much lower rates of change, were kept in a distributed file system. Distributed file systems were used effectively for Sport and Event sites with only a few thousand pages from 1996 till 2000. The requirement to deliver tens and thousands of pages in less than thirty seconds or better to large number of servers could not be met by distributed file system, however.

Figure 6 shows the architecture of the Sydney publishing system, showing the interaction of the content management system and the content distribution system. (Much of the complexity of the system has been removed from this diagram.) The complete content management system (everything above the dotted line) was duplicated to avoid any single point of failure. The HTOD was used everywhere. Not shown in the diagram are the HTOD repositories used to maintain the queues of Web objects for the distribution Trigger Monitor, Sydney, Midwest, East Coast, and West Coast hubs. The final distribution point for the Web objects was the local file system of the Web serving nodes (Ws). The Web server infrastructure in the United States is shown below the dashed line. Notice the shift between Nagano and Sydney from many expensive, replicated engines at every site to a single, large, centralized multiprocessor engine for content management and many inexpensive Web server accelerators to serve the site. Note also that in Sydney there was only a single instance of any database, finally eliminating all replications.

Figure 7 shows the Sydney 2000 Web site Australian Country page. It illustrates the depth of the information available to the web site users. If all one was interested in was how athletes from Australia were doing you could bookmark this one page. All results were available for every sport and event that Australians participated even if they finished in fourteenth place. The same result fragments was used on the athlete's pages. Available thru the Athlete pages were all the preliminary heats, games, statistics, etc. that athlete has participated in. All of this information typically was available in less than one minute after the result system triggered a new result.

At the end of the event, the fragment repository contained 277K fragments occupying almost 2G of disk space. The site finished the event with over 107K published Web objects (78K pages and 29K images). On the peak day, the content management and content distribution system processed 291K Web objects (nearly five times Nagano). 600K total trigger transactions were processed during the sixteen days creating 3.6M new and changed Web objects, which occupied 125G of data.

More than seven terabytes of data was distributed by the content distribution system to the fifty-seven nodes. Scoreboard updates were delivered from Sydney to the East Coast serving nodes in less than 6 seconds. Given automatic delays built into most television broadcast of 30 seconds, users discovered that the site had scores on the Web before the broadcasters. Even the complex result updates with hundreds of pages changed were delivered on average in less than 40 seconds to the serving nodes.


Sydney 2000
IBM
Worldwide Partner
Click here for the Sydney time
OFFICIAL SITE OF THE SYDNEY 2000 OLYMPIC GAMES - 15 SEPTEMBER TO 1 OCTOBER
Tickets | Schedule | Français
Home
Every Sport
Every Athlete
Every Country
About the Games
Sydney
Kids
Store
Venues
Paralympics
Chat
FanM@il
looksmart
Search ☑ olympics.com □ the web for
Search
The Sydney 2000 Olympic Games.
country links
Home
Participants
Medal Winners
Sport
Geography
click here for this country's complete results by sport

Australia

Canoe/kayak — sprint: Silver to Australia
Australia won silver behind Hungary's Zoltan Kammerer and Botond Storcz in the men's K2 500m canoe/kayak, Sunday. > more

Canoe/kayak — sprint: Bronze to Borchert
Australia's Katrin Borchert won bronze behind Atlanta Olympic bronze medallist Josefa Idem Guerrini in difficult windy conditions in the K1 500m final Sunday. > more

Gold medal winners Zoltan Kammerer and Botond Storcz

Latest Results > Complete results
Last Updated on 1 Oct 07:19 PM Sydney Time

Women's Modern Pentathlon After Running
Results for 1 Oct 16:20 Official

Rk	Athlete	Country	Total Points
1	COOK Stephanie	GBR	5,318
2	deRIEL Emily	USA	5,310
3	ALLENBY Kate	GBR	5,273
14	CHILLER Kitty	AUS	4,886

Women's Canoe/Kayak Sprint K2 500m Final Race
Results for 1 Oct 16:15 Official

Rk	Name	Country	Time	Time Behind
1	FISCHER Birgit WAGNER Katrin	GER	1:56.996	
2	KOVACS Katalin SZABO Szilvia	HUN	1:58.580	1.584
3	PASTUSZKA Aneta SOKOLOWSKA Beata	POL	1:58.784	1.788
6	BORCHERT Katrin WOOD Anna	AUS	2:01.472	4.476

top 5 medal standings
Last Updated 1 Oct 03:53 PM
Australia

Sport	●	●	●	Total
SW	5	9	4	18
SA	2	1	1	4
CT	1	2	3	6
AT	1	2	0	3
SH	1	1	1	3

> Country Medal Standings
> Olympic Medal Standings

headlines
- > Serial "pest" fined for disrupting marathon
- > "Best ever" Games bow out in style
- > Canoe/Kayak - Sprint: Hungary rides waves to gold
- > Canoe/Kayak - Sprint: Italy's Idem Guerrini conquers wind for gold
- > Basketball: Oh, we meant US *women's* basketball

> Australia News Archive

anthems
> Listen to the national anthem

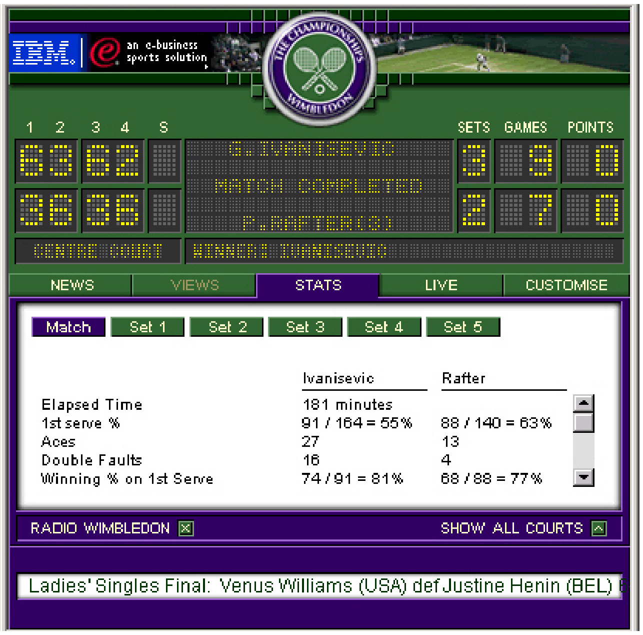
Figure 7. Sydney 2000 Australia Country Page

It should be no surprise that every event record was broken again. The traffic to the site broke the most hits on a sport event site in a single minute based on officially audited figure of 1.2M hits. Total traffic for the event over the sixteen days was 11.3 billion hits. The Web server accelerators handled less than half (5.5 billion) of the hits; the ISP proxy caches and the user browsers handled the rest (due to the expiration policy). The Web server accelerators had 80-90% cache hit ratio during the entire event. Thirty percent of the log entries on the serving nodes were UC.GIF records. Other significant statistics include the peak traffic day of 875M hits, 2.8M visits, and 18M page views and a total for the event of 35M official visits and 222M page views. Sydney was completely successfully and fulfilled the promises of being bigger than any of the previous events.

6. 2001 Wimbledon

Sports fans are a difficult audience to satisfy. For Atlanta almost all the scores were delivered after the event was over, for Nagano many events provided periodic updates, for Sydney all the major sports had sport consoles with thirty second updates. With Wimbledon 2001 [15], technology finally caught up with the fans desire for real-time scoring consoles.

With the new content distribution system, Web objects could be reliably distributed to all the geographic server sites in seconds. This enabled very short expirations of thirty seconds to be set on data for the scoreboard applets. With scoring data delivered to the most distant serving node in about six seconds and the scoring data's thirty-second expiration, the scoring console on the user's workstation is nearly always up to date. New technologies based on publish *and subscribe* were added to the event infrastructure to



1	2	3	4	5		SETS	GAMES	POINTS
6	3	6	2		G.IVANISEVIC	3	9	0
					MATCH COMPLETED			
3	6	3	6		P.RAFTER(3)	2	7	0

CENTRE COURT

WINNER: IVANISEVIC

	Ivanisevic	Rafter
Elapsed Time	181 minutes	
1st serve %	91 / 164 = 55%	88 / 140 = 63%
Aces	27	13
Double Faults	16	4
Winning % on 1st Serve	74 / 91 = 81%	68 / 88 = 77%

Ladies' Singles Final: Venus Williams (USA) def Justine Henin (BEL) 6

Figure 8. Wimbledon 2001 Tennis Console

support a real-time scoreboard. [1] Figure 8 shows a sample tennis console for Wimbledon 2001. The console is a Java Applet and uses Java Message Service (JMS) as the protocol to subscribe with the message broker. Each user acquires a permanent TCP/IP socket to a message broker. The trade off of the cost of hits to a Web server accelerator or a permanent connection to a message broker is based on update rate. Using current technology update rates of less than about thirty seconds messaging tends to be cheaper (although research into the exact "break-even" point is needed). For the 2001 Wimbledon finals between Rafter and Ivanisevic over 230K consoles were active. Two other sporting event records were broken during this event with peak hits per minute at 1.4M and peak page-views in a day at 24M.

7. 2002 Sport and Events

All the hardware and telecommunication equipment is in the process of being completely replaced. The servers are now IBM Pentium based processors running Linux. The Web server accelerators are being replaced with commercially available equipment. Multiple vendors provide telecommunication bandwidth directly (OC-12s) into each of three geographic sites. Figure 9 shows one aspect of the huge growth the sites have received. As the Web audience has increased worldwide the sites have continue to receive more hits, page-views, visits, etc. year after year.

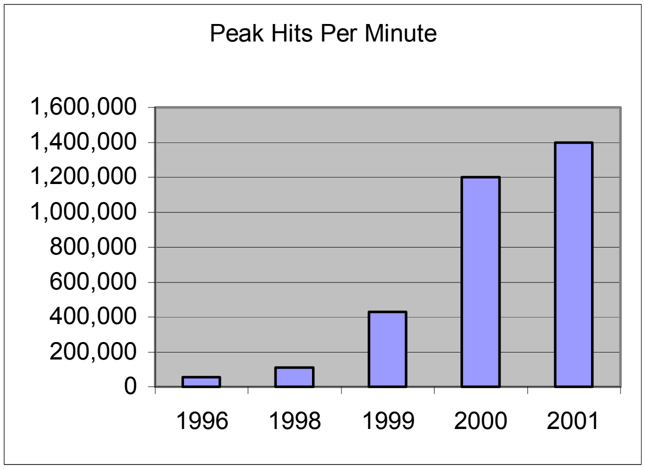


Figure 9. Sport and Events Peak Hits Per Minute

8. ACKNOWLEDGMENTS

My thanks to Jim Challenger, Arun Iyengar, Paul Reed, Cameron Ferstat, Karen Witting, Daniel Dias, and Jerry Spivak.

9. REFERENCES

[1] Banavar, G. Chandra, T. Mukherjee, B. Nagarjarao, J Strom, R.E. Sturman, D.C. An Efficient Multicast Protocol for Content-Based Publish Subscribe Systems, International Conference on Distributed Computing Systems, 1999

[2] Challenger, J. Dantzig, P.M. Iyengar, A. A Scalable and Highly Available System for Serving Dynamic Data at Frequently Accessed Web Sites, Proceedings of SC98, 1998.

[3] Challenger, J. Dantzig, P.M., Iyengar, A. A Scalable System for Consistently Caching Dynamic Web Data, Proceedings of IEEE INFOCOM'99, 1999

[4] Challenger, J. Ferstat, C. Iyengar, A., Reed, P. Witting, C. A Publishing System for Efficiently Creating Dynamic Web Content, Proceedings of IEEE INFOCOM'00, 2000

[5] Challenger, J. Iyengar, A. Jin, S. Techniques for Allocating Persistent Storage, IEEE Symposium on Large Scale Storage in the Web, April 2001.

[6] Colajanni, M. Dias, D. Yu, P. Scheduling Algorithms for Distributed Web Servers, Proceedings of International Conference on Distributed Computing Systems, 1997.

[7] Dantzig, P.D. Liu, Y. Ni, L M. Wu, C.E. A Distributed Connection Manger Interface For Web Services on IBM SP Systems, The Fourth IEEE International Conference on Parallel and Distributed Systems (ICPADS'96), 1996.

[8] Dias, D. Kish, W. Mukherjee, R. Tewari, R. A Scalable and Highly Available Web Server, Proceedings of IEEE Computer Conference, 1996.

[9] Guinness World Records, Guinness Book of Records, 1998.

[10] Java (TM Sun Microsystems) Message Service Documentation, http://java.sun.com/products/jms/docs.html

[11] 1996 Atlanta Olympic Games Web Site, July 19 – August 04, 1996, http://www.atlanta.olympic.org, site is not available.

[12] 1998 Nagano Olympic Games Web Site, February 7 – February 22, 1998, http://www.nagano.olympic.org, site is not available.

[13] 1999 The Championships Wimbledon, June 21 – July 4, 1999, http://www.wimbledon.org, site is not available.

[14] 2000 Sydney Olympic Games Web Site, September 15 – October 1, 2000, http://www.olympic.com, site is not available.

[15] 2001 The Championships Wimbledon, June 25 – July 8, 2001, http://www.wimbledon.org, site is not available.