Performance Brief

Macromedia, Inc.
600 Townsend Street
San Francisco, CA 94103
(415) 252-2000
http://www.macromedia.com

Contents

Contents 2
Executive Summary 3
  Performance Summary 3
Defining Application Performance 4
  Test Configuration and Application 4
Test Results 6
  Windows 2000 6
  Solaris 7
  Linux 9
  Performance of New ColdFusion MX Features 11
Summary 13
Appendix 14
  Test Data - Windows 2000 14
  Test Data - Solaris 14
  Test Data - Linux 14
  Test Data - Flash Remoting 15
  Test Data - CFC/UDF/Custom Tag Comparison 15
  Tack2++ Test Application Details 15

Executive Summary

ColdFusion MX delivers significant new features on an entirely new architecture from previous versions of ColdFusion Server while still delivering the same or better degree of outstanding server page processing performance and scalability across multiple CPUs under load. This document outlines Macromedia's initial findings related to these performance accomplishments.

Information is presented here comparing ColdFusion MX to the previous shipping version of the product, ColdFusion Server 5. Additionally, analysis is offered of testing performed on some of the features unique to ColdFusion MX, notably ColdFusion Components (CFCs) and Macromedia Flash Remoting.

While end-user results will vary based on application complexity and various environmental factors, users should experience comparable or improved server page processing performance and scalability over the same applications running in ColdFusion 5. Given the performance gains between ColdFusion 4.5 and ColdFusion 5, customers moving from ColdFusion 4.5 directly to ColdFusion MX are likely to experience particularly significant improvements in performance.

It would, however, be unfortunate to simply migrate existing ColdFusion applications from ColdFusion 5 to ColdFusion MX. The added benefits of the new server architecture and development architecture (CFCs) in ColdFusion MX are many and, as this document demonstrates, come at no additional cost to application performance or scalability.

Performance Summary

- Windows - An average of 10% faster than ColdFusion 5 with more linear SMP scalability
- Solaris - An average of 30% faster than ColdFusion 5
- Linux - An average of 45% faster than ColdFusion 5

Defining Application Performance

A high performing application is able to deliver content to users quickly. In this brief, application performance is analyzed by measuring page response time: the elapsed time between the submission of a request (the clicking of a submit button, the manual entry of a URL, or the clicking of a link by a user) and the successful completion of that request. A lower response time per request (completing the request more quickly) allows the application to deliver more content overall to more users in a given period of time (throughput).

There are three basic factors that influence response time performance:

- Web application server architecture and configuration
- Network infrastructure
- Web page design

All three factors are of considerable importance, and no Web application will function efficiently if any factor is neglected. However, since network bandwidth and Web page design are not functions of the web application server, this brief focuses solely on the changes in page response time that result from enhancements to ColdFusion.

Test Configuration and Application

Testing for this brief was performed with ColdFusion 5 and ColdFusion MX on single servers with varying multi-processor configurations. These were running on Microsoft Windows 2000 Advanced Server with IIS 5.0, Sun Solaris with Netscape Enterprise Server 4.0, and Red Hat Linux with Apache HTTP Server version 1.3.22. Analysis of the results is presented in the following sections grouped by operating system. Figure 1 below illustrates the basic configuration in which the tests were performed.

For each operating system, data are presented comparing the performance of ColdFusion 5 with ColdFusion MX. On Windows 2000, additional data is provided illustrating the ability of ColdFusion MX to provide linear SMP scalability as the number of processors increases resulting in higher throughput.

Testing of ColdFusion Components for scalability and performance in relation to other methods of code encapsulation and testing of Macromedia Flash Remoting scalability were performed on a single Microsoft Windows 2000 server with varying multi-processor configurations.

To achieve optimal performance, tests were conducted with adjustment to the ColdFusion Server settings for simultaneous requests. This setting indicates the number of simultaneous requests that ColdFusion will serve at any given time. When the server reaches the specified limit, requests are queued and handled in the order received. The general rule is to set the simultaneous requests setting to a value 3-4 times the number of CPUs on the system. For example, if the server has two processors, the number of simultaneous requests for the ColdFusion server should be set to 6-8. In this case, if and when there are more than six threads being requested, ColdFusion automatically queues additional requests until one of the other threads completes processing. Note that the optimal number may vary depending on hardware configuration and the application being run. Real-world deployments of ColdFusion applications will benefit from the configuration of each application's particular optimal simultaneous request setting.

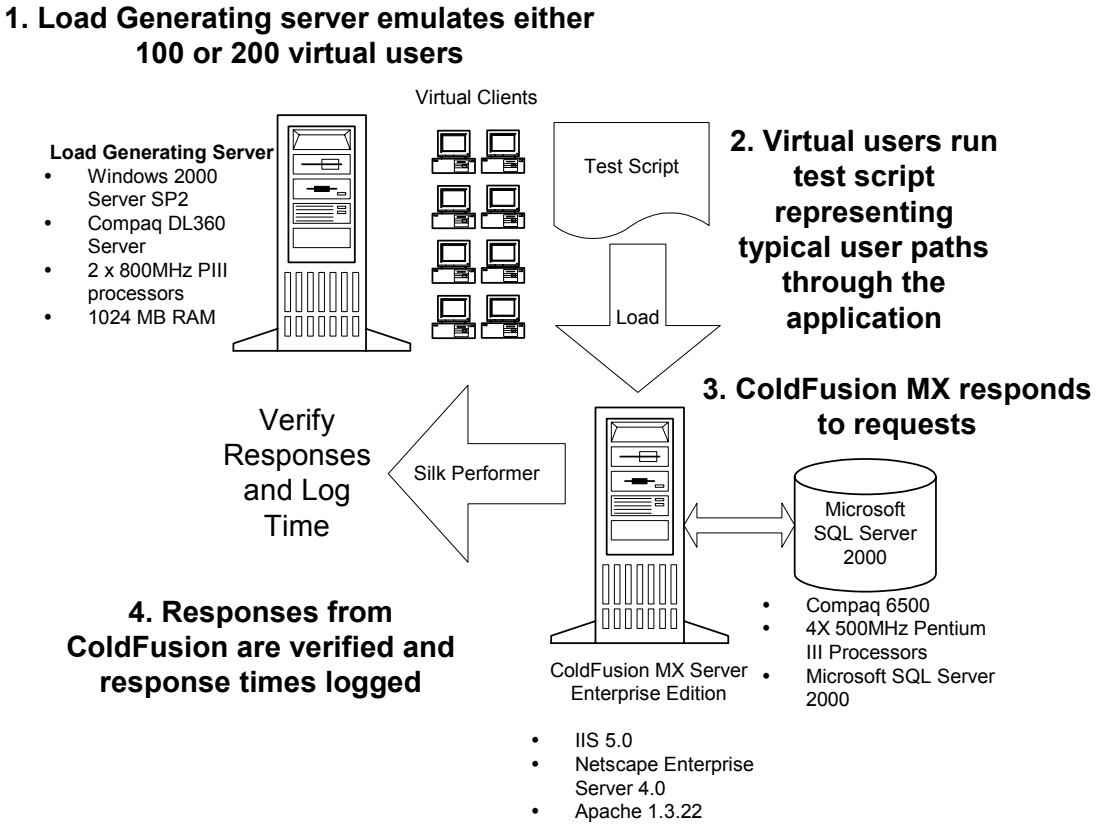

Figure 1: The basic testing configuration used for this brief

All tests were also conducted with the ColdFusion "trusted cache" setting enabled. This specifies that any requested files found to currently reside in the template cache will not be inspected for potential updates. For sites where templates are not updated during the life of the server, this minimizes file system overhead. This is an important configuration setting to ColdFusion MX performance in general and should be used appropriately in all application deployments.

Testing for this brief was performed in Macromedia's Enterprise Testing facility in Newton Massachusetts. The application used in the test is the Tack2++ sample application, a new version of the original Tack2 Online Store sample application included with ColdFusion MX. It preserves all of the functional characteristics of the original Tack2Plus application but makes use of template and query caching features, as would a real-world production ColdFusion application. The application represents a typical e-commerce Web application found on the Web. For more details on the application, see the Appendix.

Test Results

Windows 2000

Two different server configurations were used for Windows 2000 testing.

SMP Scalability Testing:

- Windows 2000 Server SP2
- Intel OCPRF100 Server
- 2,4, and 8 x 500MHz PIII Xeon processors
- 4 GB RAM

ColdFusion 5 and ColdFusion MX Performance Comparison Testing:

- Windows 2000 Server SP2
- Compaq 1850 Server
- 2 x 500MHz PIII Xeon processors
- 512 MB RAM

The back-end database for each test suite was a separate Compaq 6500 with four 500MHz Pentium III Processors running Microsoft SQL Server 2000. All tests were performed using 100 virtual users (VUs) simulated with Segue SilkPerformer V.

SMP Scalability Results

To demonstrate linear SMP scalability with ColdFusion MX, the processor configuration of the test machine was gradually increased from one to four processors. As illustrated in Figure 2, the average page response time for the application improved in a linear fashion as processors were added. Additionally, this demonstration of scalability was accompanied by performance results slightly above those achieved by ColdFusion 5 in the same tests.

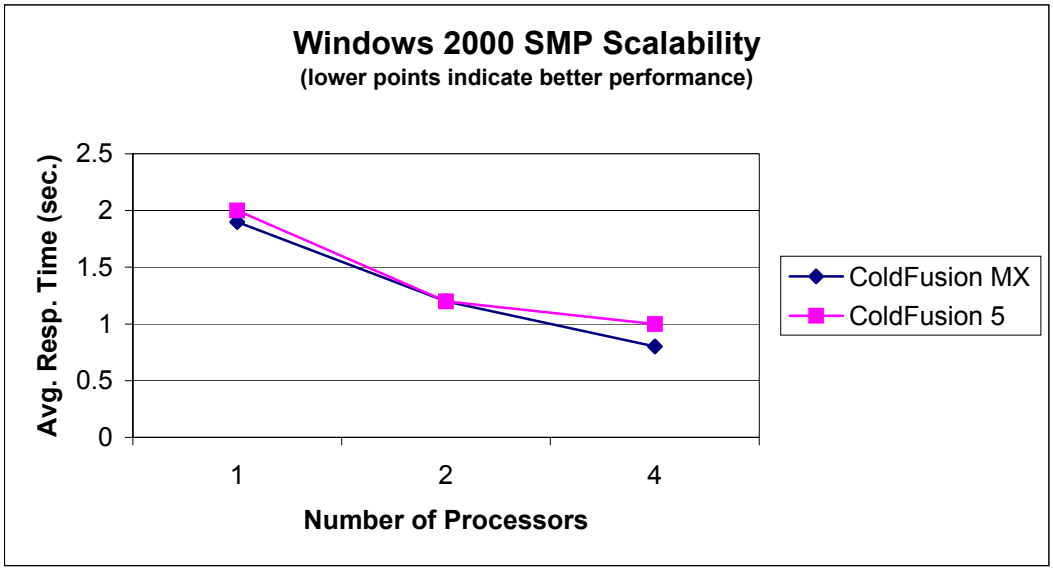

Figure 2: Linear improvement in average page response time with addition of processors on Windows 2000. Tests were conducted with 100 VUs and illustrate near linear scalability in SMP configurations with ColdFusion MX.

With the ability to process page requests more quickly, applications deployed on ColdFusion MX are able to service more requests than the same applications deployed on ColdFusion 5 (as illustrated in Figure 3).

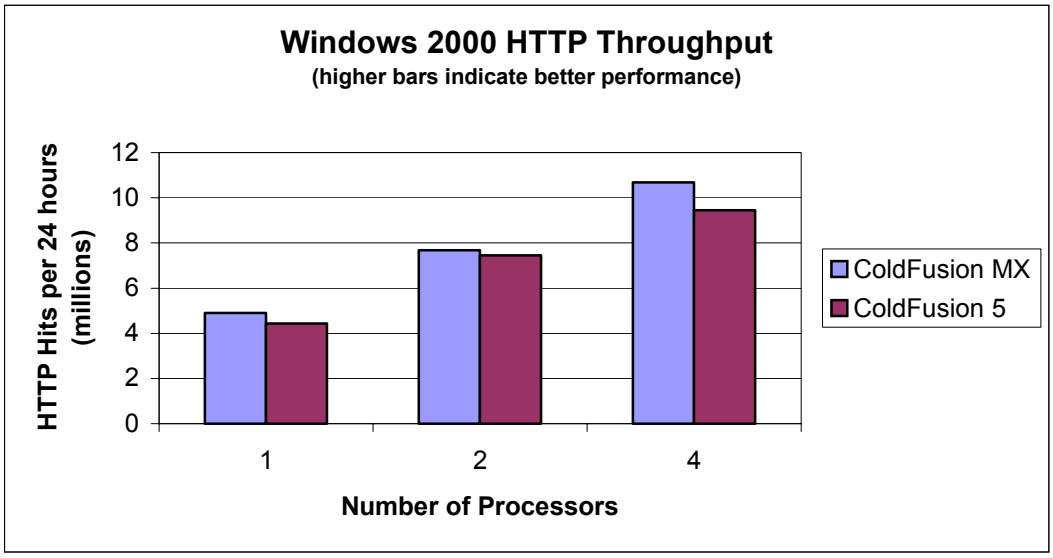

Figure 3: HTTP requests served over a 24 hour period on Windows 2000.

Tests were conducted using 100 virtual users (VUs) on identical hardware configurations. For this test, loading of images was disabled and no static HTML or HTML framesets were used. Therefore, HTTP requests in this case are equal to the number of requests made. See the Appendix for a detailed explanation of the test application and HTTP request data.

Both ColdFusion 5 and ColdFusion MX perform and scale exceptionally well. ColdFusion MX does, however, have some other performance impacting variables to consider. For example, these tests were performed with the Java Virtual Machine (JVM) that ships with the product (Sun 1.3.1_02). There are alternative JVMs available, and conducting these tests with a different JVM may produce quite different results. There is no comparable performance variable in ColdFusion 5.

In addition, each application is unique and achieving optimum performance will require slightly different configurations from application to application. Some experimentation in a testing environment will allow this ideal configuration to be identified. The ColdFusion MX Application Development Center frequently publishes articles and tutorials providing insight into ColdFusion development and server configuration best practices (http://www.macromedia.com/desdev/mx/coldfusion/).

Solaris

The performance comparison between ColdFusion 5 and ColdFusion MX on Solaris was conducted on a server with the following specifications:

- Solaris 7
- Sun E220 Server
- 2 x 450MHz SparcV9 processors

- 512 MB RAM

The back-end database for each test suite was a separate Compaq 6500 with four 500MHz Pentium III Processors running Microsoft SQL Server 2000. All tests were performed using 100 virtual users (VUs) simulated with Segue SilkPerformer V.

As with Windows 2000, the ability to process page requests more quickly allows ColdFusion MX to service more requests than ColdFusion 5 during an equivalent period of time. Figure 4 illustrates an improvement in ColdFusion MX of nearly 30% over ColdFusion 5.

*Note: Data for scalability across different processor configurations for Solaris and Linux was not available at the time when this brief was published. Dual processor servers were used for all tests on both operating systems.

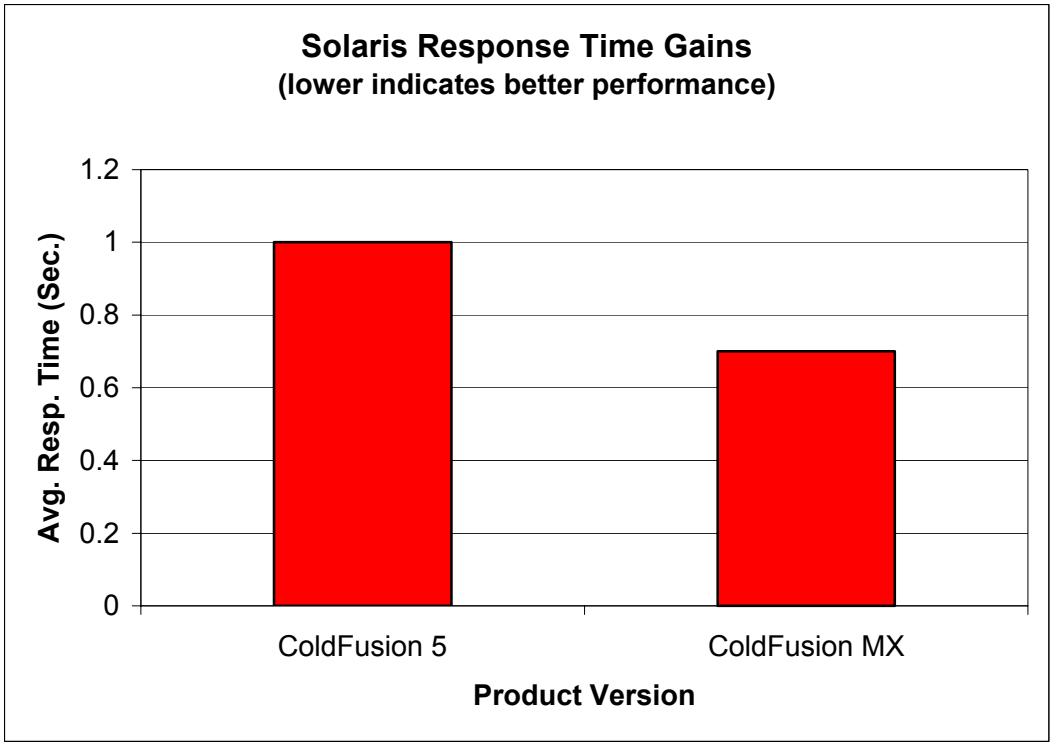

Figure 4: Reduction in average page response time from ColdFusion 5 to ColdFusion MX on dual processor Solaris server.

The ability of ColdFusion MX to handle more responses results in throughput of more content over a 24 hour period, as illustrated by Figure 5.

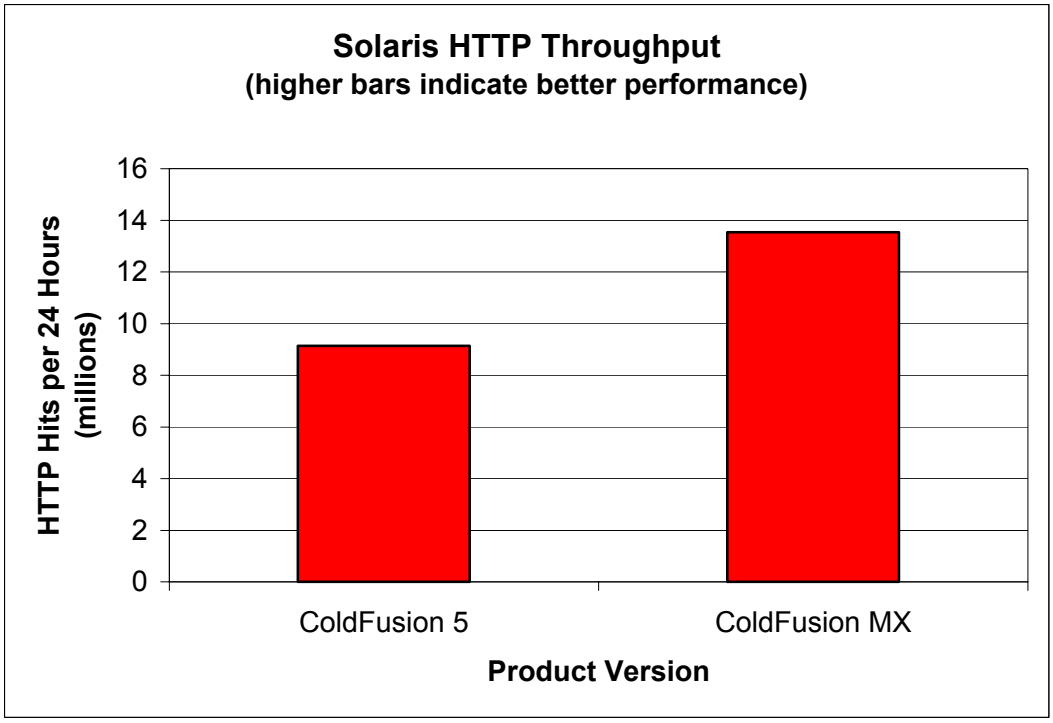

Figure 5: Increased volume of HTTP throughput over a 24-hour period with ColdFusion MX compared to ColdFusion 5.

Linux

The performance comparison between ColdFusion 5 and ColdFusion MX on Linux was conducted on a server with the following specifications:

- RedHat 7.2
- Penguin Computing server
- 2 x 933MHz PIII Processors
- 512 MB RAM

The back-end database for each test suite was a separate Compaq 6500 with four 500MHz Pentium III Processors running Microsoft SQL Server 2000. All tests were performed using 100 virtual users (VUs) simulated with Segue SilkPerformer V.

This test showed ColdFusion MX nearly twice as fast as ColdFusion 5.

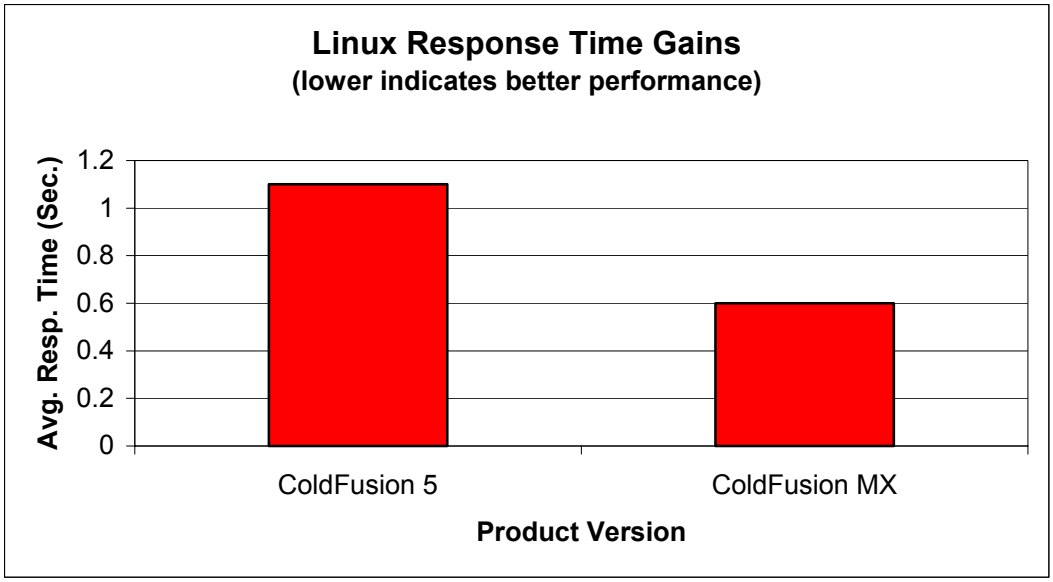

Figure 6: Running on Linux, ColdFusion MX served pages in approximately half the average page response time of ColdFusion 5.

This improvement would result in nearly doubling the amount of content that could be served from the same hardware running ColdFusion MX (see Figure 7).

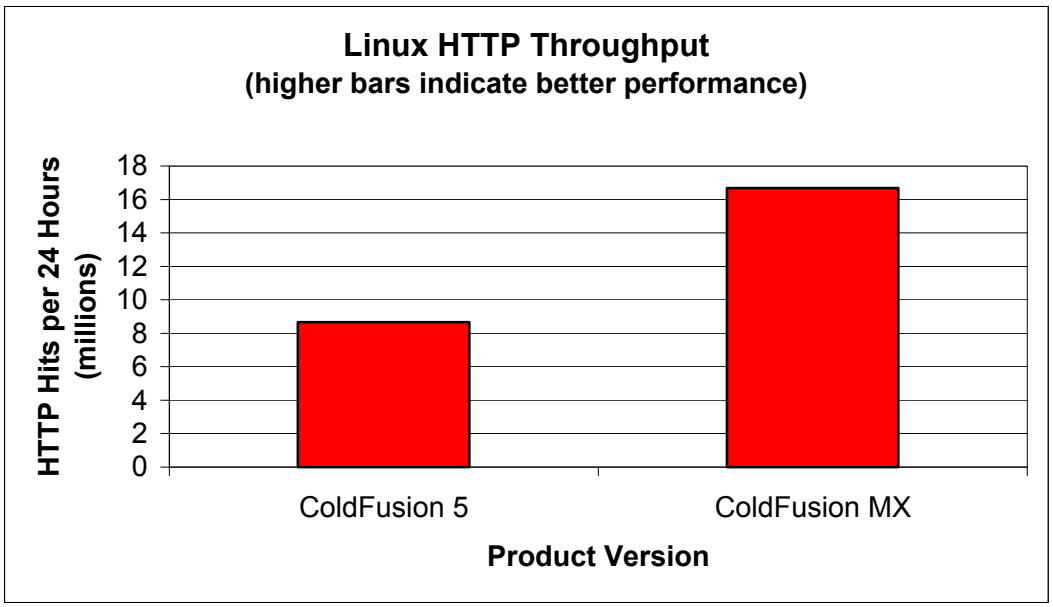

Figure 7: HTTP throughput gains with ColdFusion MX on Linux.

Performance of New ColdFusion MX Features

Two new features in ColdFusion MX, ColdFusion Components (CFCs) and Flash Remoting, are expected to play a significant role in applications deployed with this product. ColdFusion Components provide an approachable way to build structured ColdFusion applications, thereby increasing reuse and simplifying maintenance. Macromedia Flash Remoting enables tight integration between ColdFusion applications running on the server and rich Internet clients built with Macromedia Flash MX technology.

Since both features area available only with ColdFusion MX, direct comparisons to ColdFusion 5 are not possible. However, the following data demonstrate how ColdFusion Components compare to previously available methods for performing similar tasks and shows how Flash Remoting applications calling CFCs are able to scale in multi-processor configurations.

ColdFusion Component Comparison Testing Results

ColdFusion Components encapsulate an application's functionality and make it available to a wide variety of consumers, including Web browser based application, other ColdFusion developers, Macromedia Flash Applications, and SOAP Web Services. While no previous technique existed in ColdFusion CFML to do exactly the same thing as can be done with CFCs, there are other methods of encapsulating code for reuse, notably user defined functions and custom tags.

This test endeavored to illustrate that applications can take advantage of the benefits of using ColdFusion Components over other code reuse and encapsulation methods *without any reduction in application performance*. The test code was based on a user-defined function published by the Common Function Library Project (http://www.cflib.org) that provides date-time conversion from one time zone to another (see appendix for details about each version of the test code).

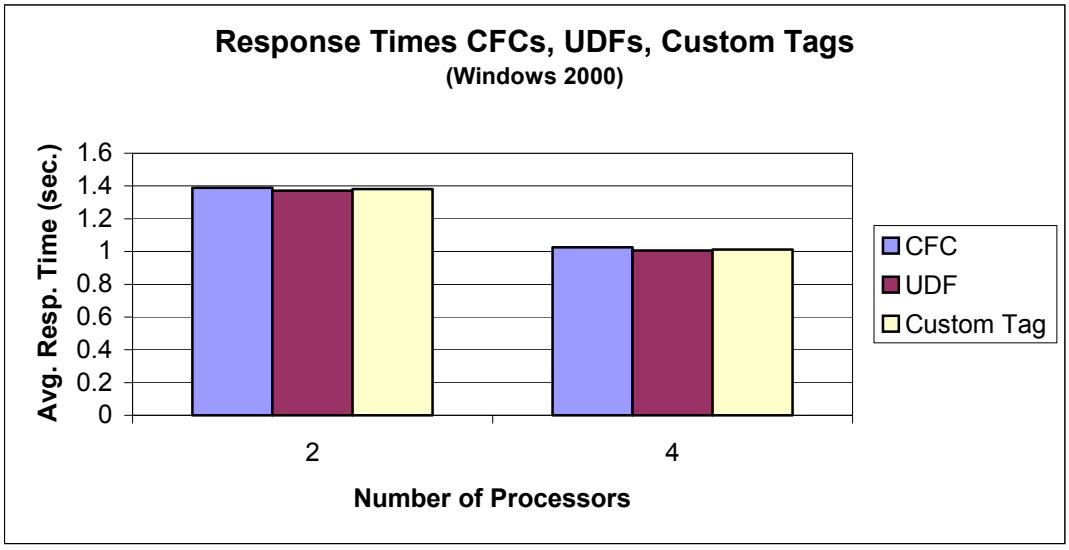

Figure 8: New capabilities of ColdFusion Components perform without any significant reduction in application performance compared to comparable techniques for encapsulating application logic.

Figure 8 illustrates nearly identical performance of all three techniques on both dual and quad-processor servers running ColdFusion MX. This illustrates that customers can take advantage of the powerful new capabilities of ColdFusion Components, thereby reducing development and maintenance time, *without experiencing any processing overhead*.

ColdFusion Components in Flash Remoting Test Results

Many applications are now being deployed using Macromedia Flash for high-impact user interfaces. Macromedia Flash MX Remoting services are an integrated feature of ColdFusion MX and allow Flash to natively access ColdFusion Components for server-side logic. Testing indicates that CFCs being accessed via Flash Remoting were able to improve data throughput in a near-linear fashion as processors were added to the ColdFusion server (see Figure 9).

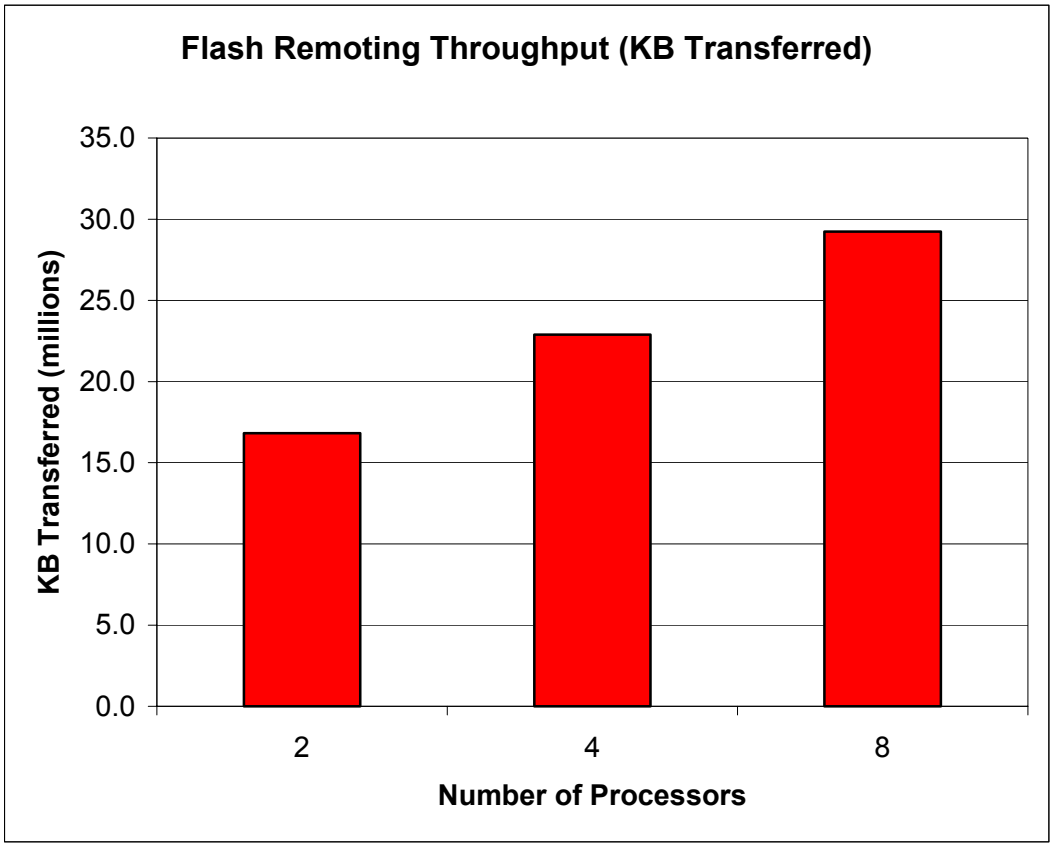

Figure 9: Results of Flash Remoting SMP scalability testing

To present the type of server side functionality that would be more commonly found in Flash Remoting applications, the above CFC-UDF-Custom Tag comparison code was not used. Rather, a CFC with logic from the core Tack 2++ application was used. See appendix for more information about the test code.

Both the CFC-UDF-Custom Tag comparison and Flash Remoting tests were performed with Segue SilkPerformer V. However, unlike the previous operating system specific performance and scalability sections of this paper, they were both conducted with 200 virtual users (VU). Consequently, ColdFusion Components and Flash Remoting accessed components actually handled a more difficult load in these tests.

Summary

ColdFusion MX performs as well or better than ColdFusion 5 and, considering that ColdFusion 5 was already as much as five times faster that its predecessor (ColdFusion 4.5.1 SP2), this is quite an accomplishment. Yet ColdFusion MX is much more than simply a performance enhanced version of ColdFusion 5. Breakthrough new capabilities like ColdFusion Components, Flash Remoting, SOAP Web Services support, XML processing capabilities and much more are really the key strength of this release. As the data in this performance brief show, all of these revolutionary new capabilities come at no increased cost to the excellent performance and scalability of your applications.

Appendix

Test Data - Windows 2000

Response Time Data: 100 Virtual Users (VU)

# Processors	1	1	2	2	4	4
Server Version	CF 5	CFMX	CF 5	CFMX	CF 5	CFMX
Simultaneous Requests	4	3	8	6	12	12
Avg. Response Time (sec.)	2	1.9	1.2	1.2	1	0.8

Throughput Data: 100 Virtual Users (VU)

# Processors	1	1	2	2	4	4
Server Version	CF 5	CFMX	CF 5	CFMX	CF 5	CFMX
HTTP Hits/Min	3,080.40	3,406	5,174.40	5,334	6,564	7,417
HTTP Hits/Hour	184,824	204,372	310,464	320,040	393,840	445,032
HTTP Hits/Hour X24	4,435,776	4,904,928	7,451,136	7,680,960	9,452,160	10,680,768

Test Data - Solaris

Response Time Data: 100 Virtual Users (VU)

Server Version	CF5	CFMX
Simultaneous Requests	8	6
Avg. Response Time (sec.)	1	0.7

Throughput Data: 100 Virtual Users (VU)

Server Version	CF 5	CFMX
HTTP Hits/Min	6,350.40	9,400.20
HTTP Hits/Hour	381,024	564,012
HTTP Hits/Hour X24	9,144,576	13,536,288

Test Data - Linux

Response Time Data: 100 Virtual Users (VU)

Server Version	CF 5	CFMX
Simultaneous Requests	4	8
Avg. Response Time (sec.)	1.1	0.6

Throughput Data: 100 Virtual Users (VU)

Server Version	CF 5	CFMX
HTTP Hits/Min	6,021.60	11,587.80
HTTP Hits/Hour	361,296	695,268
HTTP Hits/Hour X24	8,671,104	16,686,432

Test Data - Flash Remoting

Response Time Data: 200 Virtual Users (VU)

# Processors	2	4	8
Avg. Response Time (sec.)	0.8	0.5	0.4

Throughput Data: 200 Virtual Users (VU)

# Processors	2	4	8
KB Data Transferred/Min	11,674.8	15,894.6	20,301.6
KB Data Transferred/Hour	700,488.0	953,676.0	1,218,096.0
KB Data Transferred/Hour X 24	16,811,712.0	22,888,224.0	29,234,304.0

Test Data - CFC/UDF/Custom Tag Comparison

Response Time Data: 200 Virtual Users (VU)

# Processors	2	4
Avg. Response Time (sec.)	1.4	1.0

Throughput Data: 200 Virtual Users (VU)

# Processors	2	4
HTTP Hits/Min	8,682.60	11,820.00
HTTP Hits/Hour	520,956	709,200
HTTP Hits/Hour X24	12,502,944	17,020,800

Tack2++ Test Application Details

The Tack2++ application represents a typical e-commerce application consisting of a catalog, shopping cart, check out pages, etc. While actual performance gains experienced with other applications will vary depending on scope and complexity, customers can expect to realize similar results with similar applications and server configurations.

The user load mix used in this round of testing is generally more demanding than is typically experienced on public e-commerce sites. In the test suite, each virtual user initiates and completes a purchase, whereas studies have shown that approximately 90% of all traffic on a typical e-commerce site consists of browsing activities, and less than 5% of all users actually add items to the shopping cart and check out.

Each user session in the test suite consisted of the following types of requests:

1. Main Menu
2. Show Category
3. Show Items
4. Show Specific Item
5. Add Item to Cart
6. Check-Out
7. Customer Information Entry

8. Order Confirmation

Simultaneous HTTP requests were generated to simulate 100 virtual client sessions. This simulated activity at thresholds of between approximately 3 to 12 million requests per day. As explained above, variations in the number of requests per day between ColdFusion Server 5 and ColdFusion MX Server are a result of different server settings for the number of simultaneous requests allowed. Each stress test was performed multiple times to confirm accuracy. Averaged performance numbers were used for this document.

In real-world Web applications, most of the session time is spent waiting for the user to do something. A large Web site or application may have hundreds of open user sessions, while only a few dozen are actively making requests to the application server. The rest of the sessions are waiting for the user to do something.

Recreating realistic user pauses and mistakes is very difficult, so for the purposes of this test, user pauses were removed altogether. Thus, 100 simultaneous virtual user sessions does not represent 100 simultaneous users. Rather, the number of simultaneous users would be at least an order of magnitude greater. While the exact ratio of open sessions to active sessions is difficult to generalize because of the disparity in Web applications, a test using 500 virtual users would roughly translate to 10,000 open user sessions.