



# Adobe® ColdFusion® 8 performance brief

The fastest version yet, Adobe ColdFusion 8 enables developers to build and deploy Internet applications and web services that perform with exceptional speed and stability.

## Table of contents

Executive summary.....	1
Statistics summary .....	2
Performance Enhancements in ColdFusion 8 .....	3
Overall Server Performance.....	3
Specific Area Performance.....	4
CFC (ColdFusion Component) Creation and Invocation .....	4
CFSwitch/CFCase.....	5
List Manipulation.....	5
Adobe ColdFusion 8 Standard vs Enterprise Performance.....	6
New Performance-Focused Features of ColdFusion 8.....	6
CFThread.....	6
FileFunctions.....	7
Impact of JDK on ColdFusion 8 Performance .....	8
Summary.....	8
APPENDIX .....	9
Defining application performance .....	9
Test Applications and Scripts .....	9
Test Configuration.....	9
ColdFusion Server Settings.....	10
Test Methodology.....	11
Additional Performance Tuning provided by Adobe ColdFusion 8.....	11
Reproducing the results .....	12

## Executive summary

Adobe® ColdFusion® 8 software strengthens the proven performance of Macromedia® ColdFusion MX 7 to deliver applications capable of processing significantly more page requests than applications from any previous versions of ColdFusion. ColdFusion 8 has new features such as CFThread, file-functions, and Stored Procedure Caching that enable users to develop better performing applications. Administrators can also tune ColdFusion 8 for increased performance with additional server monitoring and profiling controls such as Disable CFC type checking and request controls.

Overall Server Performance of Adobe ColdFusion 8 is up to 4.25 times faster than Macromedia ColdFusion MX 7.

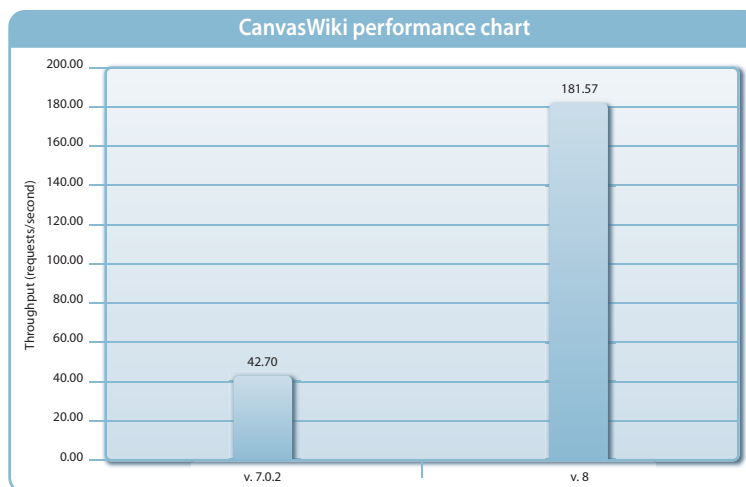


Figure 1. shows Canvas Wiki is 4.25 times faster in ColdFusion 8 compared with ColdFusion MX 7.0.2.

Performance improvement in specific functional areas of ColdFusion 8 is up to 38.7 times faster than ColdFusion MX 7.0.2.

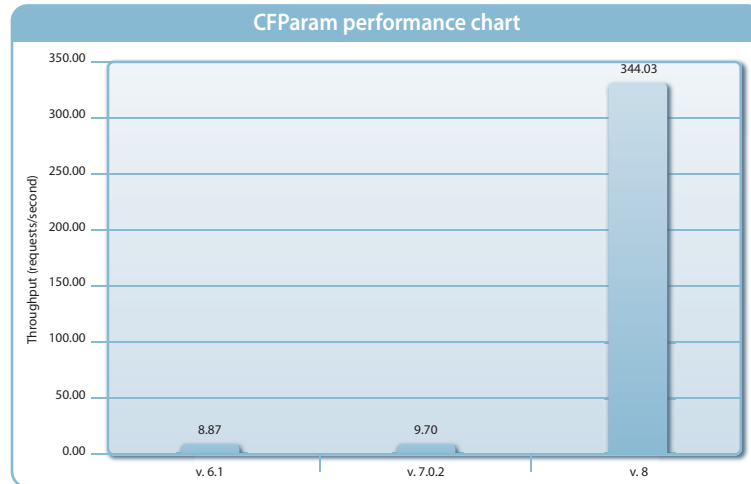


Figure 2. shows CFParam is 38.7 times faster in ColdFusion 8 than ColdFusion MX 7.0.2 and 35.4 times faster than ColdFusion MX 6.1.

### Statistics Summary

Following sections outline the performance gains demonstrated by ColdFusion 8 over ColdFusion MX 7.0.2 and ColdFusion MX 6.1 by category.

#### Overall server performance

Area	Improvement over v. 7.0.2	Improvement over v. 6.1
Overall server performance (CanvasWiki)	4.25X	NA
Overall server performance (BlogCFC)	48%	45%

Table 1. Overall server performance gains in ColdFusion 8 compared to ColdFusion MX 7.0.2 and ColdFusion MX 6.1.

#### Specific area performance

Area	Improvement over v. 7.0.2	Improvement over v. 6.1
CFParam	38.7X	35.4X
CFC creation and invocation	23.1X	7.1X
REReplaceNoCase	7.5X	7.5X
CFForm	7%	NA
Date functions	6X	6X
Evaluate	2.9X	2.8X
List functions	3X	3X
CFSwitch/CFCase	2.7X	4.3X
REFindNoCase	2.7X	2.7X
ColdFusion structures	1.9X	1.9X
IsDefined	2X	10%

Table 2. ColdFusion 8 performance gains in specific areas ColdFusion 8 compared with ColdFusion MX 7.0.2 and ColdFusion MX 6.1.

## Performance enhancements in ColdFusion 8

Extensive internal performance enhancements were made with ColdFusion 8. The entire ColdFusion MX 7.0.2 code base was analyzed by running real-world applications on special ColdFusion MX 7.0.2 installs built with instrumentation. Nearly 100 customer applications consisting of approximately 2.4 million lines of ColdFusion Markup Language (CFML) code were used to analyze CFML usage patterns and find bottlenecks in real-world customer settings.

When issues arose, the server runtime code was modified to eliminate them. As each bottleneck was uncovered and removed, each change was individually measured to gauge its performance impact on the server. Most changes were small, resulting in incremental improvements as small as 1–2%, but their cumulative effect is substantial.

Specific areas such as ColdFusion components (CFC) and structures were analyzed for performance bottlenecks and tuned for optimum performance. Substantial performance gains were obtained in these areas (as shown in Table 2), which further contributed to the overall server performance.

### Overall server performance

We measured gains in application runtime performance by comparing throughput in a test application on ColdFusion MX 6.1, ColdFusion MX 7.0.2, and ColdFusion 8. Throughput is defined as the number of requests from the server per second. To compare overall server performance, we used CanvasWiki and BlogCFC. These applications represent typical content-centric web applications found on the Internet or intranets that extensively use CFML tags and CFCs. While actual performance gains by other applications will vary depending on scope and complexity, it is important to note that the BlogCFC and CanvasWiki applications were not modified specifically for this white paper. As a result, real-life customer applications may or may not realize comparable results by using similar applications and server configurations. The testing methodology, details of the test configuration, and test applications are available in the appendix.

When CanvasWiki was used with ColdFusion 8, it performed 4.25 times faster than when used with ColdFusion MX 7.0.2. CanvasWiki is not supported by ColdFusion MX 6.1.

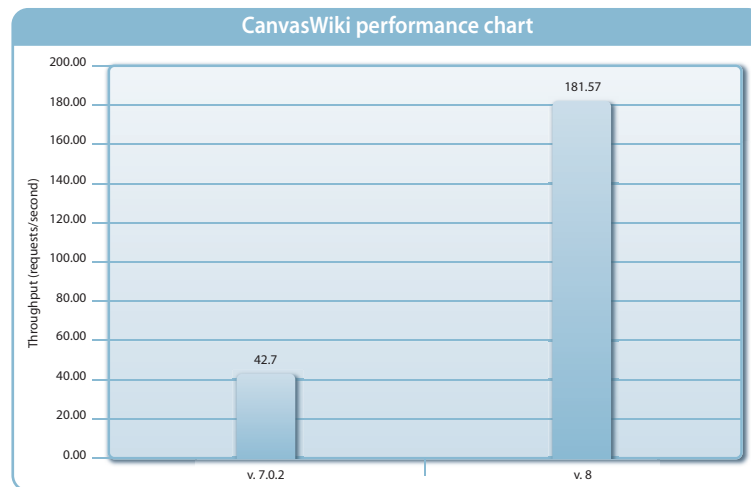


Figure 3. shows 4.25X performance gain on ColdFusion MX 7.0.2 in CanvasWiki application.

When the BlogCFC application was used with ColdFusion 8, it performed 48% faster than ColdFusion MX 7.0.2 and 45% faster than ColdFusion MX 6.1.

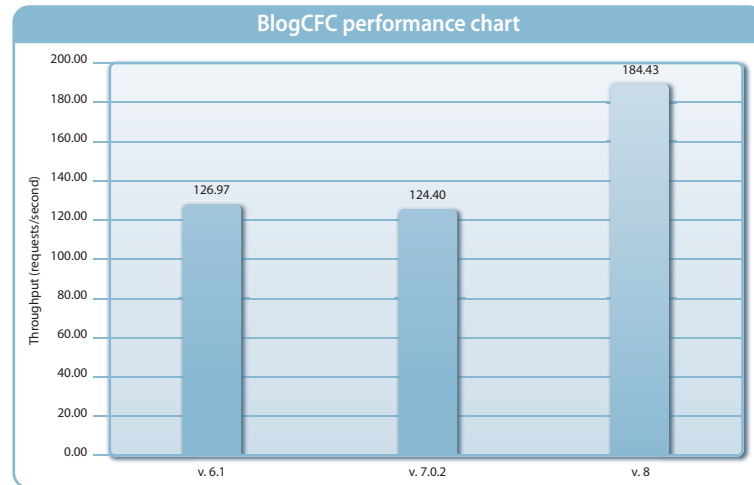


Figure 4. shows 48% performance gain on ColdFusion MX 7.0.2 and 45% on ColdFusion MX 6.1 in BlogCFC application.

### Specific area performance

#### CFC (ColdFusion Component) creation and invocation

ColdFusion 8 dramatically improves the performance of CFC, a core ColdFusion feature. CFC creation performance is 23 times faster in Adobe ColdFusion 8 compared with ColdFusion MX 7.0.2 and 7.1 times faster when compared with ColdFusion MX 6.1. Details about the script and source code used in these measurements can be found in the appendix.

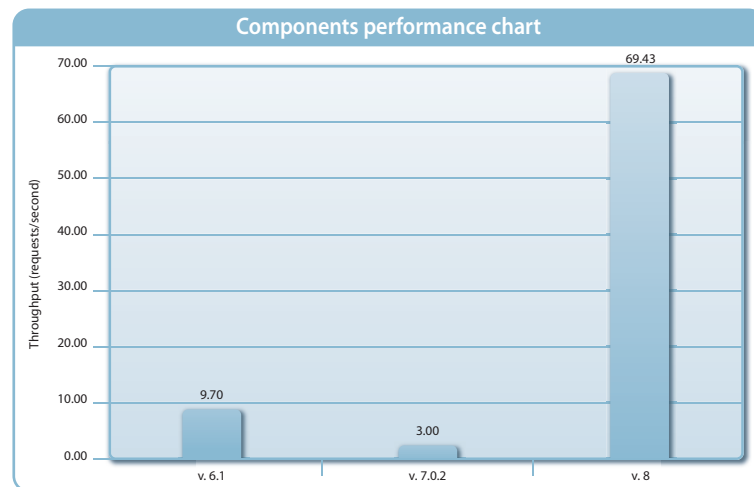


Figure 5. shows performance gains of 23.1X in Adobe ColdFusion 8 as compared to ColdFusion MX 7.0.2 and 7.1X as compared to ColdFusion MX 6.1.

### CFSwitch/CFCase

CFSwitch/CFCase is one of the most widely used tags in ColdFusion. In ColdFusion 8, it performed 2.7 times faster than ColdFusion MX 7.0.2 and 4.3 times faster than ColdFusion MX 6.1.

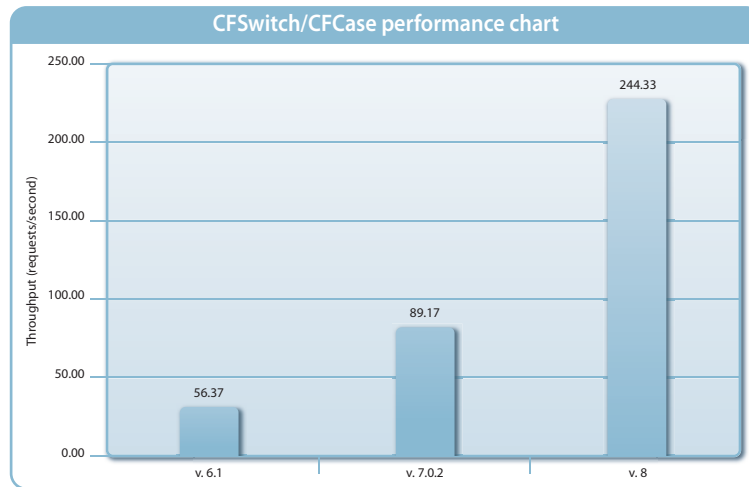


Figure 6. shows CFSwitch/CFCase is 2.7 times faster in ColdFusion 8 than ColdFusion MX 7.0.2 and 4.3 times faster than ColdFusion MX 6.1.

### List manipulation

The lists and list manipulation functions, which are widely used in ColdFusion applications, perform significantly faster in ColdFusion 8 than in previous versions of ColdFusion. We used a ColdFusion template that created lists and used the list manipulation functions for testing. In ColdFusion 8, list functions are three times faster than in ColdFusion MX 7.0.2 and ColdFusion MX 6.1.

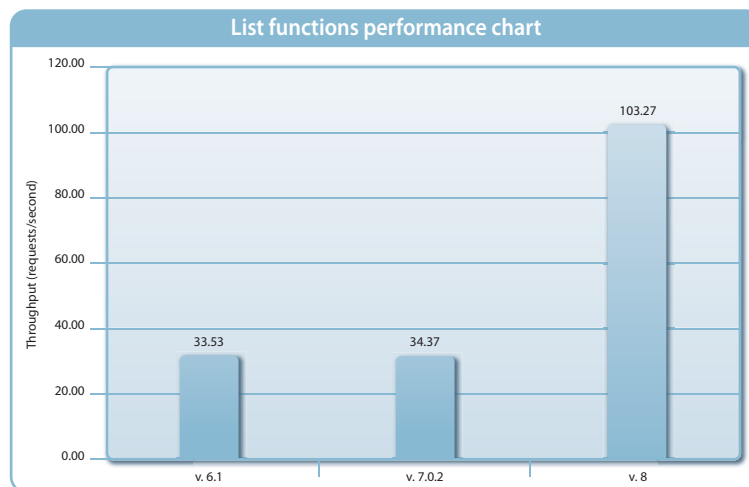


Figure 7. List Functions are 3 times faster in ColdFusion 8 than ColdFusion MX 7.0.2 and ColdFusion MX 6.1.

## Performance of ColdFusion 8 Standard Edition vs. Enterprise Edition

ColdFusion 8 is available in two editions: Standard and Enterprise. PDF document generation in ColdFusion 8 Enterprise Edition uses enhanced tuning and a multithreaded architecture to deliver linear scalability under load to accommodate high-volume document generation applications. ColdFusion 8 Enterprise Edition scales better under load than Standard Edition.

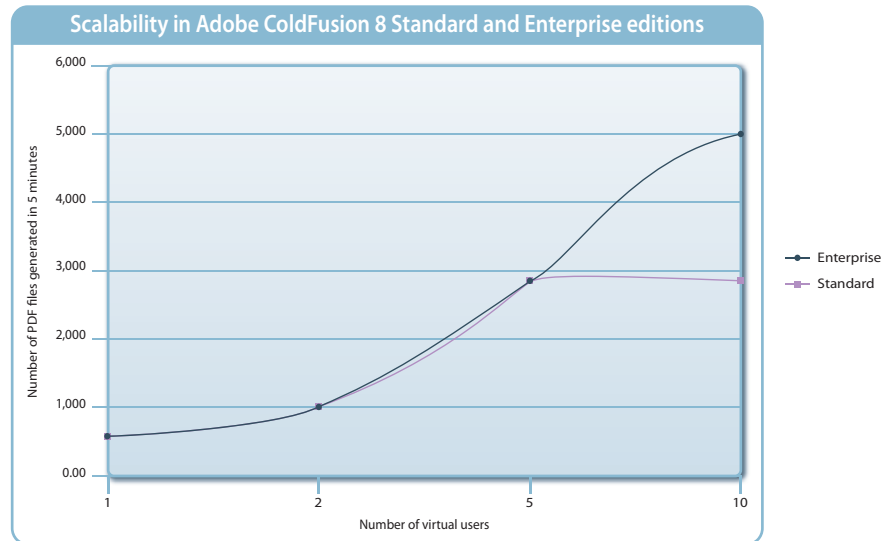


Figure 8. shows CF8 is scales better in Enterprise version than Standard.

The appendix lists the scripts and administrator settings used for this test. ColdFusion 8 Enterprise Edition also delivers linear scalability under load and enhanced tuning for the applications listed in Table 3.

Document services	Advanced protocol support
PDF form processing	Gateway architecture
PDF document manipulation	Enterprise IM gateways (sametime, XMPP)
Server-side printing	Asynchronous CFML gateway
Presentation generation	SMS gateway
PDF generation	JMS gateway
FlashPaper® generation	TCP/IP socket gateway
Structured reporting	Directory watcher gateway
Exchange server integration	
Multithreaded CFML (CFTHREAD)	
Dynamic e-mail	

Table 3. Types of applications that scale better in ColdFusion 8 Enterprise Edition compared with ColdFusion 8 Standard Edition.

## New performance features

### CFThread

ColdFusion 8 introduces a new CFML-based tag that enables application developers to quickly and easily add powerful multithreading capabilities to server applications with the new, CFThread tag. This enables asynchronous processing in CFML, which harnesses the power of today's processors to vastly improve overall user response times where long-running tasks are made up of autonomous processing steps and processed synchronously today.

This powerful capability has a significant effect on user response time for web applications, especially Adobe Flex™ and Ajax applications where many server-side requests are made from a browser and server response delays can make an application appear unresponsive.

An example scenario where asynchronous multithreading may be applied is illustrated in Figure 10:

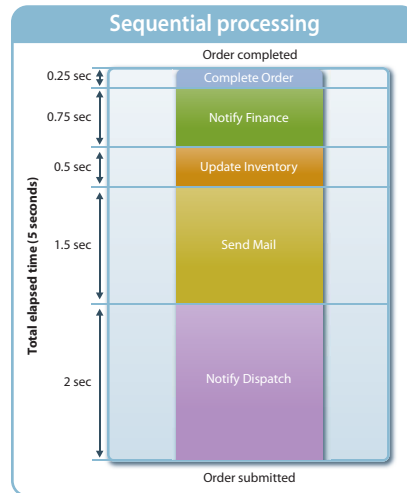


Figure 10. shows various tasks in a sample sequential order processing workflow.

In this example, total elapsed process response time can be reduced from five seconds to two seconds, since all steps can be performed in parallel using separate threads and the new CFThread tag.

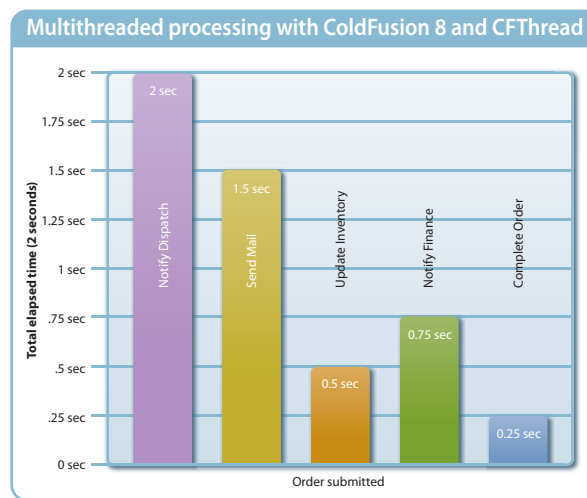


Figure 11. shows the performance improvement, from 5 seconds to 2 seconds total elapsed time, if this task is achieved using CFThread to perform the asynchronous tasks when compared to if the tasks are performed sequentially.

### File functions

File functions are introduced for the first time in ColdFusion 8 to solve the performance issue with CFFile reading large files.

The graph shows the performance improvement in terms of the number of files processed across file sizes by using new file processing capabilities available with ColdFusion 8 compared with those available in ColdFusion MX 7.

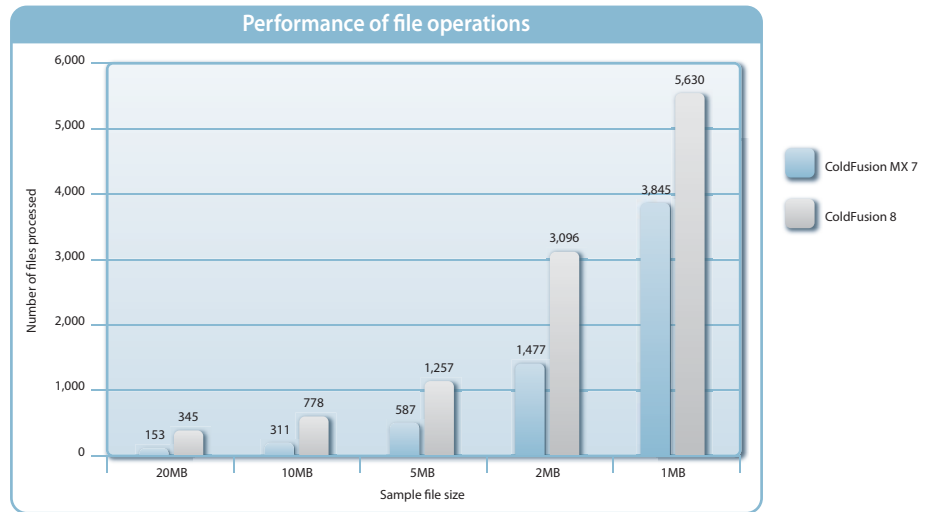


Figure 12. shows CF8 file functions do a faster file processing when compared to CF7.

### Impact of JDK on ColdFusion 8 performance

ColdFusion 8 ships JDK 1.6\_01 as the default JDK. However, the performance gains exhibited in our tests are marginally attributed to the JDK upgrade. The vast majority of performance gains in ColdFusion 8 are because of a substantial product tuning effort. To substantiate this assertion, we performed tests running JDK (JDK 1.4.2\_09) on ColdFusion 8, ColdFusion MX 7.0.2, and ColdFusion MX 7.

#### Specific area performance on the same JDK (JDK 1.4.2\_09)

Area	Response time v. 7	Response time v. 7.0.2	Response time v. 8 with JDK 1.4.2_09	Response time v. 8 with JDK 1.6
CreatObject in CFLOOP (1 – 10000)	1094ms	5344ms	172ms	110ms
CFParam in CFLOOP (1 – 10000)	794ms	828ms	250ms	171ms
CFSet in CFLOOP (1 – 1000000)	813ms	843ms	260ms	180ms
Array Creation in CFLOOP (1 – 100000)	125ms	130ms	60ms	33ms
Struct Creation in CFLOOP (1 – 100000)	156ms	156ms	94ms	51ms

Table 4. Performance gains in specific areas of ColdFusion 8 compared with ColdFusion 7.0.2 and ColdFusion 7 running on the same JDK.

These tests were performed by wrapping the code with GetTickCount() and averaging the results across multiple runs.

### Summary

ColdFusion 8 enables developers to build and deploy Internet applications and web services that perform with exceptional speed and stability on servers under load. Continued enhancements in runtime performance and compiler performance make ColdFusion 8 the best version of ColdFusion for developers and users.

ColdFusion 8 Enterprise Edition is an ideal solution for delivering multiple websites and applications on one or more servers or on existing J2EE application server installations. It features all the capabilities of ColdFusion 8 Standard Edition, plus special features for managing multi-application environments, asynchronous processing, and many other features that enable highly scalable and reliable implementations of important business applications.



## Appendix

### Defining application performance

A high-performing application delivers content to users quickly. In this performance brief, application performance was analyzed in terms of page response time, which is the time that elapses between a submitted request (such as clicking a submit button, manually entering a URL, or clicking a link) and completion of that request. A lower response time per request allows the application to deliver more content to more users in a given period of time (higher throughput).

Three factors have considerable impact on page response time:

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

All three factors are important and web applications will not function efficiently if any one is neglected. This brief, however, focuses solely on web application server architecture and configuration. In particular, it examines page response times in different versions of ColdFusion.

This performance brief measures gains in application runtime performance by comparing average page response times in a test application across ColdFusion 8 and earlier versions of ColdFusion. Like many large-scale content management applications, the application used in these tests makes intensive use of ColdFusion Markup Language (CFML). Applications that depend more on database transactions or I/O operations may experience less significant performance gains than the application tested for this white paper, since those factors depend less on the application server.

### Test applications and scripts

For specific area performance, a separate CFML script was created for each area. Each script covered most of the functionality in that area and were authored in such a way that more processing occurred in the script than in request processing.

The ColdFusion applications used were BlogCFC 5.8.001 (<http://blogcfc.riaforge.org/index.cfm>) and CanvasWiki 2.003 (<http://canvas.riaforge.org>). (We would like to thank Raymond Camden for giving us permission to use his applications.) Both applications represent typical content-centric web applications and make extensive use of CFML tags and CFCs. However, BlogCFC is more database-centric than CanvasWiki, and CanvasWiki is more CFC-centric than BlogCFC. While performance gains in other applications will vary depending on the scope and complexity of the application, it is important to note that BlogCFC was not tuned for this white paper. As a result, customer applications with similar server configurations may not realize results presented in this performance brief.

### Test configuration

ColdFusion 8 is available in Standard Edition and Enterprise Edition and may be deployed in several configurations, depending on edition, application needs, and server environment.

ColdFusion Enterprise Edition may be deployed in one of three configurations: standalone or server, multiserver, or J2EE. In the server configuration, the underlying J2EE application server, JRun™, is not exposed to the user or application administrator. The multiserver configuration supports multiple virtual instances of ColdFusion running on a single physical server providing increased availability, application isolation, improved application-level configuration, and clustering. In the J2EE configuration, ColdFusion 8 is deployed on an existing installation of a standard J2EE application server, such as IBM WebSphere or BEA WebLogic. We have used the standalone configuration for testing.

ColdFusion MX 6.1, ColdFusion MX 7.0.2, and ColdFusion 8 were set up to use IIS Web Server 6.0, and each was set up on a different website. During the collection of results, the appropriate website was turned on, and others were turned off.

The test applications and the individual performance scripts were run with ColdFusion MX 6.1, ColdFusion MX 7.0.2, and ColdFusion 8 (Build 173894) on Microsoft® Windows® 2003 with the following specifications:

- Windows 2003 Enterprise Edition with Service Pack 2
- Microsoft Internet Information Server 6.0
- HP ProLiant DL360 G5
- Two 2GHz Intel® Xeon® Dual-Core 5130 processors
- 4GB of RAM

The back-end database was on a separate system with the following specifications:

- Windows 2003 Enterprise Edition with Service Pack 2
- Microsoft Internet Information Server 6.0
- SQL Server 2005
- HP ProLiant DL360 G5
- Two 2GHz Intel Xeon Dual-Core 5130 processors with 1333MHz front-side bus
- 2GB of RAM

The client was on a separate system with the following specifications:

- Windows 2003 Enterprise Edition with Service Pack 2
- HP ProLiant DL360 G5
- Two 2GHz Intel Xeon Dual-Core 5130 processors with 1333MHz front-side bus
- 2GB of RAM

The paging file size on the test systems was increased to 4GB. Visual settings were set for Best Performance instead of Best Appearance.

#### ColdFusion server settings

ColdFusion MX 6.1	ColdFusion 8
RDS disabled	RDS disabled
Trusted cache enabled	CF documentation not installed
Enabled white-space management	Trusted cache enabled
CF documentation not installed	Enable white-space management
Max number of JRun threads: 20	Max number of JRun threads: 50
Max Flash® Remoting requests: NA	Max Flash Remoting requests: 1
Max web service requests: NA	Max web service requests: 1
Max CFC function requests: NA	Max CFC function requests: 1
Max number of simultaneous templates: 20	Max number of simultaneous templates: 20
xmx: 512M	xmx: 512M

ColdFusion MX 7
RDS disabled
CF documentation not installed
Trusted cache enabled
Enable white-space management
Max number of JRun threads: 20
Max Flash Remoting requests: NA
Max web service requests: NA
Max CFC function requests: NA
Max number of simultaneous templates: 20
xmx: 512M

### Deviations from ColdFusion server settings

For Application testing, Scalability and File Functions

<b>ColdFusion MX 7.0.2</b>
Max number of simultaneous templates: 10
Max number of JRun Threads: 10
<b>ColdFusion 8</b>
Max number of simultaneous templates: 10
Max number of JRun threads: 10
<b>ColdFusion MX 6.1</b>
Max number of simultaneous templates: 10
Max number of JRun threads: 10

For CFThread testing

<b>ColdFusion MX 7.0.2</b>
Max number of simultaneous templates: 10
Max number of JRun threads: 10
<b>ColdFusion 8</b>
Max number of simultaneous templates: 10
Max number of JRun threads: 113
Max number of threads for CFThread: 100
<b>ColdFusion MX 6.1</b>
Max number of simultaneous templates: 10
Max number of JRun threads: 10

### Test methodology

For testing applications, simultaneous HTTP requests were generated to simulate 30 no-think-time virtual users. This simulated activity tested thresholds between approximately 2 to 15 million requests per day. Each stress test was performed multiple times to confirm the accuracy of the results. The results were averaged and presented in this document.

In real-world web applications, most session time is spent waiting for the user to do something. This is particularly true for content-centric websites, where a large portion of time is spent reading web pages. While a large website or application may have hundreds of open user sessions, only a few dozen of these sessions are actively submitting requests to the application server.

It is difficult to realistically simulate user pauses and mistakes in software testing and thus user pauses were not included in this testing. As a result, 30 simultaneous virtual user sessions do not represent 30 simultaneous users. While it is difficult to generalize the exact ratio of open sessions to active sessions because of the differences among web applications, a test using 30 virtual users most likely translates to approximately 600 open user sessions.

### Additional performance tuning provided by ColdFusion 8

Additional performance tuning parameters provided by ColdFusion 8 can be tuned to achieve optimum performance. Server monitoring and profiling helps ColdFusion developers and users identify performance bottlenecks in their applications.

#### Server Monitor

The ColdFusion 8 Server Monitor enables you to track activities on a ColdFusion server. You can identify information about the server including requests, queries, memory usage, and errors. You can also start and stop collecting server information and take snapshots of the server. For more information about server monitoring, refer to ColdFusion 8 documentation.

#### Disable CFC type check

This tuning control is available in the ColdFusion Administrator under the settings section. When checked, UDF arguments of the CFC type are not validated, resulting in an increase of performance. The arguments are treated as type "ANY". Use this setting in a production environment only.

#### Request tuning

Request tuning is a new section in the ColdFusion 8 administrator under Server Settings. The following request types control the number of simultaneous requests that can be processed.

- CFM page requests
- CFC method invocations (via an HTTP GET)
- Flash Remoting requests (old and new)
- Web service requests

These settings are contained within the ColdFusion 8 code and work across all J2EE application servers.

Additionally, special provisions have been made to URLs beginning with /CFIDE. These requests will not be restricted the same as 'normal' requests are and will always be let through. This allows administrators to log in to the ColdFusion Administrator, and run the Server Monitor when the server is heavily loaded. Note: If no requests are being processed and the J2EE server is queuing requests without passing the request along to the ColdFusion 8 servlets, administrators will not be able to run the Server Monitor or Administrator.

Administrators do not need to restart ColdFusion 8 for configuration changes to take effect. The following are usage scenarios for request tuning:

- **Scenario A**—A server is under severe normal load from CFM page requests and is virtually unresponsive. An administrator runs the server monitoring tool to find out which pages are being requested. Since the Flash Remoting requests are not subject to the same throttle as the CFM page requests, the administrator collects information about the heavily loaded server.
- **Scenario B**—A user deploys an Ajax-based application using the new ColdFusion 8 Ajax features. His application URL is posted on SlashDot, and a large number of people use his application, causing thousands of CFC invocations to be made against the server. Since the CFC invocations and the CFM requests have a different pool, other CFM-based applications will still run with reasonable throughput.

#### **Stored procedure caching**

CachedWithin and CachedAfter have been added to the cfstoredproc tag and can be used to cache the stored procedure results. Refer to the documentation on cfstoredproc for additional information.

#### **Reproducing the results**

To reproduce the results mentioned in this document, unzip CF8PerfTest.zip from the location [http://download.macromedia.com/pub/coldfusion/8/performance/cf8\\_perftest.zip](http://download.macromedia.com/pub/coldfusion/8/performance/cf8_perftest.zip) and follow the instructions in the Readme.txt file.

