

Improve application performance and scalability with Adobe® ColdFusion® 9

Table of contents

- 1: Executive summary
- 2: Statistics summary
- 3: Existing features
- 7: New features
- 11: Summary
- 12: Appendix: Testing details and additional product information

Executive summary

Performance tests reveal that Adobe ColdFusion 9 software provides significant performance improvements over ColdFusion 8, delivering applications capable of processing considerably more page requests. ColdFusion 9 also offers an array of new features such as page fragment caching, object caching, implicit getter/setter in ColdFusion components (CFCs), directory functions, and others that enable developers to build higher performing applications. Additionally, administrators can tune ColdFusion 9 for increased performance with the new Cache Template in Request setting.

Overall server performance in ColdFusion 9 is up to 40% faster than in ColdFusion 8 and up to 6 times faster than ColdFusion MX 7.

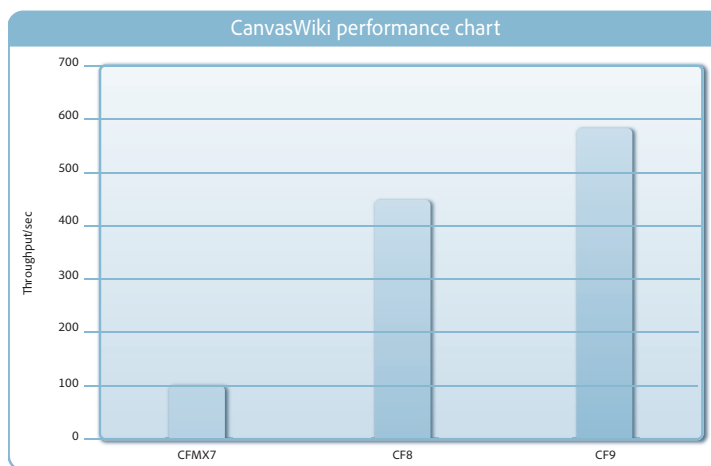


Figure 1. CanvasWiki is 40% faster in ColdFusion 9 than in ColdFusion 8 and 6 times faster than in ColdFusion MX 7.

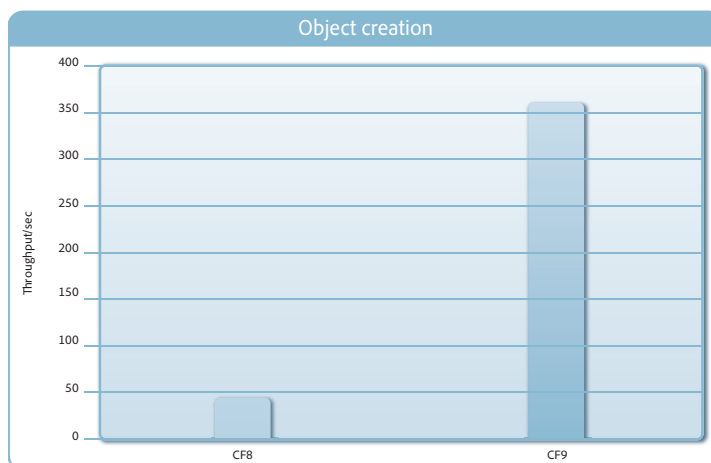


Figure 2. CFC object creation in ColdFusion 9 is 8 times faster than in ColdFusion 8.

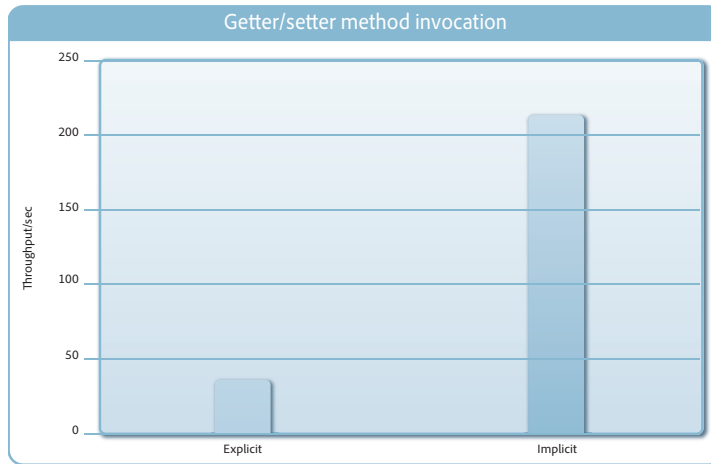


Figure 3. The implicit getter/setter method invocation is 7 times faster than the explicit getter/setter invocation.

Statistics summary

Performance enhancements in ColdFusion 9

Through extensive research, bottlenecks in ColdFusion 8 were identified, analyzed, and removed to achieve significant performance gains and preserve backward compatibility in ColdFusion 9. Substantial gains were obtained in areas such as CFC object creation/invoke, CreateUUID, evaluate, and Flash® Remoting. Table 1 shows the overall server performance gains in ColdFusion 9 compared with ColdFusion 8 and ColdFusion MX 7. Table 2 shows a summary of all the performance improvements in the existing features.

New features such as advanced caching, virtual file system, and a set of directory functions deliver additional performance gains.

Overall server performance		
Area	Improvement over ColdFusion 8	Improvement over ColdFusion MX 7
Overall server performance (CanvasWiki)	40%	500%

Table 1. Overall performance gains in ColdFusion 9 compared with ColdFusion 8 and ColdFusion MX 7.

Existing feature performance	
Area	Improvement over ColdFusion 8
CFC object creation	700%
CFC method invocation	200%
Flash Remoting	800%
Page caching	30%
UUID key generation	5,300%
Date functions	35%
Evaluate	30%
Performance improvement with IIS	100%

Table 2. Performance improvements in existing features.

New feature performance		
	Disabled	Enabled
Cache Template in Request	x	50x

New feature performance		
	<cfdirectory>	DirectoryList()
Nonrecursive DirectoryList()	x	10x
Recursive DirectoryList()	x	4x

New feature performance		
Getter/setter	Explicit	Implicit (added by ColdFusion 9)
Object creation	x	4x
Method invocation	x	7x

Table 3. Performance improvements over ColdFusion 8 if new features are used.

Existing features

ColdFusion 9 provides significantly better performance in several feature areas, including object creation, method invocations, and many frequently used functions.

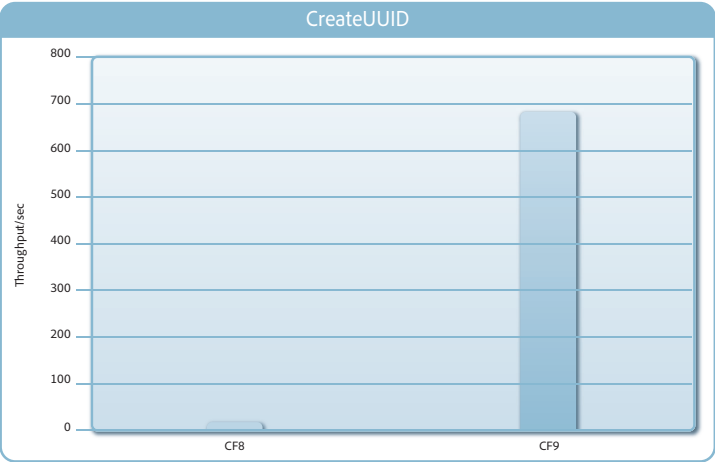


Figure 4. Creating UUIDs in ColdFusion 9 is 53 times faster than in ColdFusion 8.

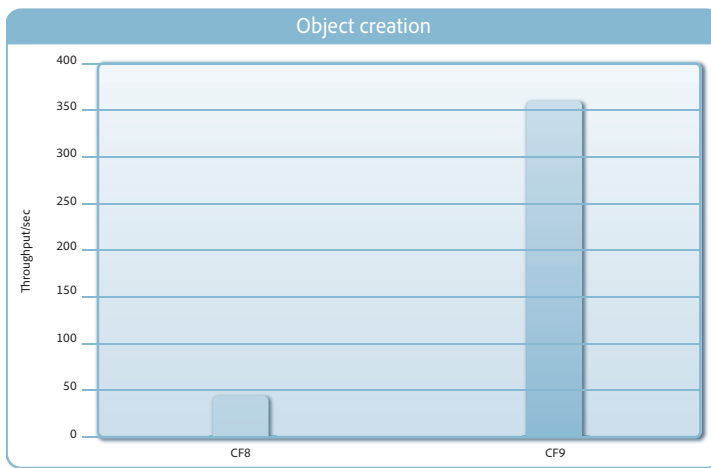


Figure 5. CFC object creation in ColdFusion 9 is 8 times faster than in ColdFusion 8.

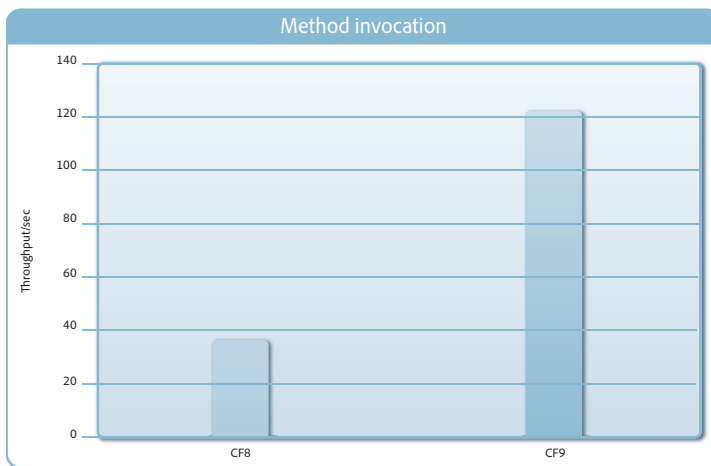


Figure 6. Method invocation in a ColdFusion 9 CFC is 3 times faster than in ColdFusion 8.

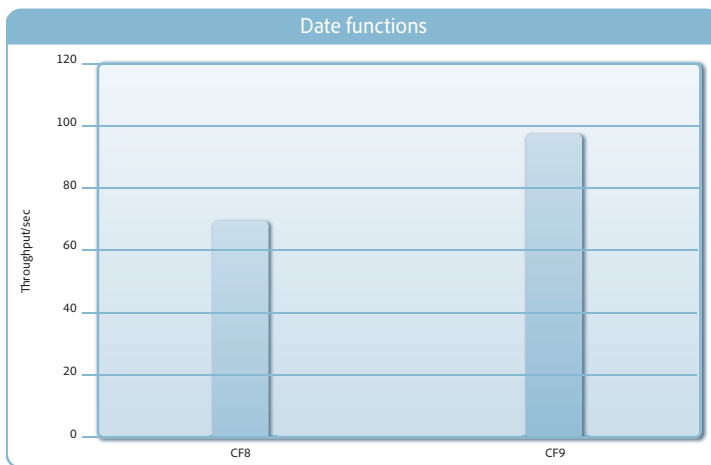


Figure 7. Date functions show a 35% performance improvement in ColdFusion 9.

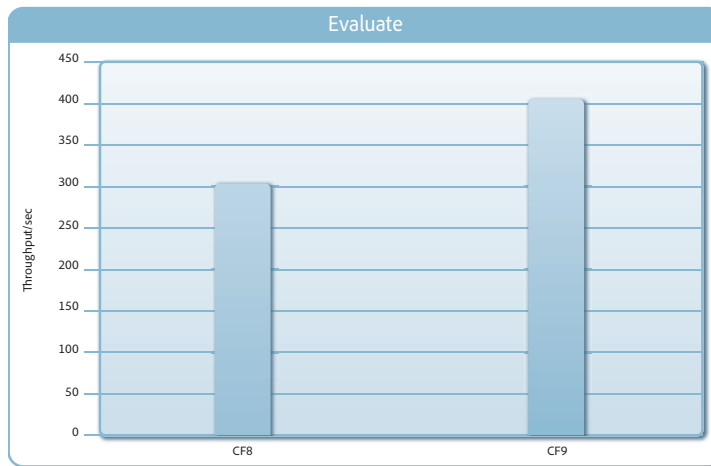


Figure 8. The evaluate function shows a 30% performance improvement in ColdFusion 9.

IIS connector

The Microsoft Internet Information Services (IIS) connector shows performance improvements if there are requests for static contents. Figure 9 graphs the performance for 50 virtual users simultaneously requesting 10 ColdFusion Markup Language (CFML) files and 10 static HTML pages. As requests for static content increase, performance improves. Performance gains are visible only if there is enough load to consume all IIS worker threads. For more information, refer to "IIS connector improvements" in the appendix.

Note: This performance gain is available only with IIS 6 and IIS 7 on Microsoft® Windows® 2003 and later.

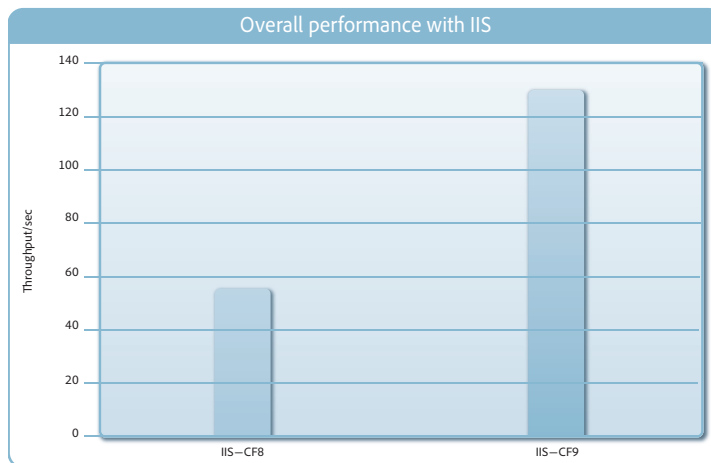


Figure 9. IIS connectors show a 100% performance hike in ColdFusion 9.

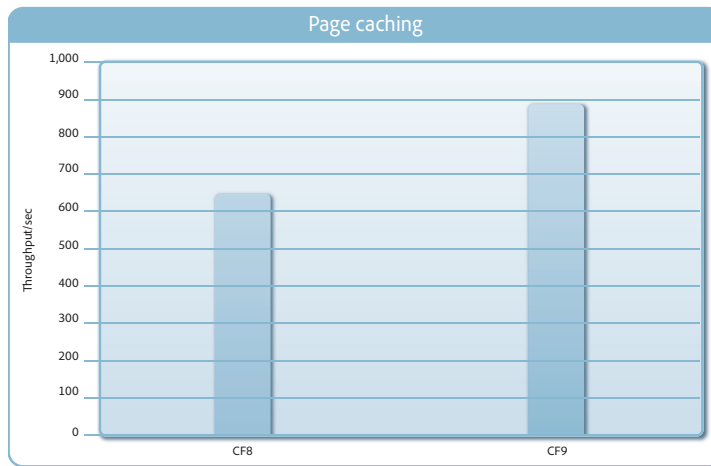


Figure 10. Page caching improves 30% in ColdFusion 9.

Flash Remoting

ColdFusion 9 provides enhanced Flash Remoting performance apart from bug fixes for "circular reference serialization-deserialization."

From an Adobe Flex® application, 10,000 to 50,000 ActionScript® objects are sent to the ColdFusion 9 server. The AS objects contain the data types most commonly used in ActionScript. These objects are serialised and deserialised into CFCs in the ColdFusion 9 server and then sent back to the Flex application. This turnaround process is defined as a server trip. The throughput shown in Figure 11 is the average of 10 such server trips for each object.

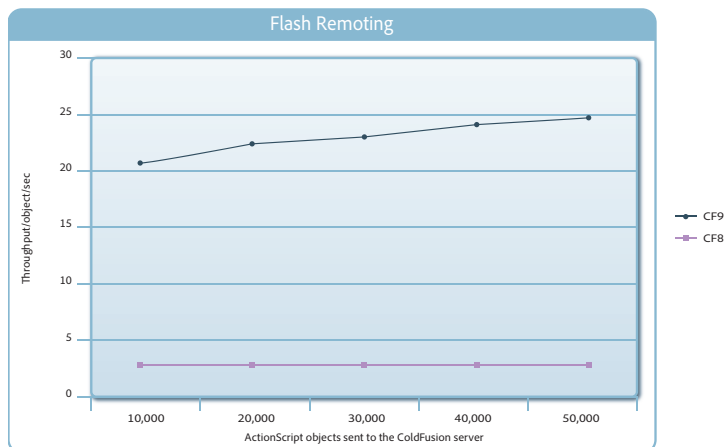


Figure 11. Flash Remoting in ColdFusion 9 is 9 times faster than in ColdFusion 8.

New features

ColdFusion 9 introduces several new features such as advanced caching and implicit getters and setters that enable developers to build higher performing applications.

Advanced caching—object caching

ColdFusion 9 allows users to cache specific objects using `cachePut()` and to retrieve objects with `cacheGet()`. Users also have the option to put, get, and flush cached objects.

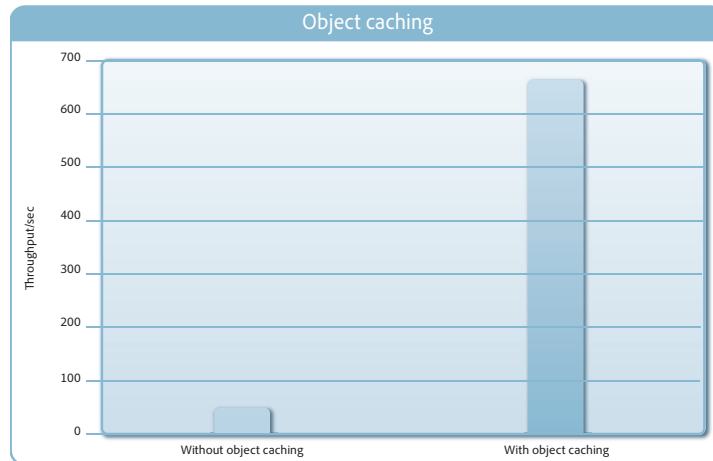


Figure 12. ColdFusion 9 improves performance when a 500k image is read and resized as a cached object.

Advanced caching—page fragment caching

In many applications, ColdFusion pages contain a combination of static and dynamic information that changes at different frequencies. In these cases, it would be ideal to improve application performance while still allowing application users access to information that is frequently updated. ColdFusion 9 allows developers to cache only the portions of the page that change infrequently or not at all, enabling them to fine-tune application performance.

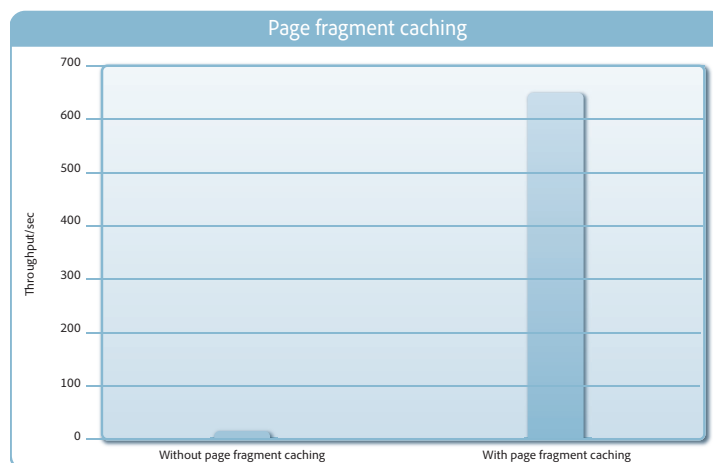


Figure 13. When invoking a web service that converts Celsius to Fahrenheit, page fragment caching dramatically improves performance.

Cache template in request

ColdFusion 9 provides a new Cache Template in Request setting. When this setting is enabled, ColdFusion will not inspect requested templates in the current request for potential updates. This setting can be enabled in the ColdFusion Administrator on the caching page under Server Settings.

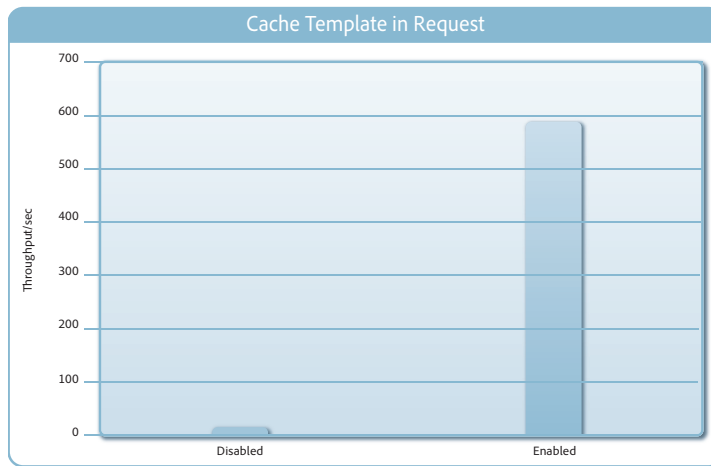


Figure 14. Cache Template in Request is a setting in the ColdFusion 9 Administrator that improves performance by 50 times.

DirectoryList() function

ColdFusion 9 introduces a new function, `DirectoryList()`, that returns a list of the contents in the directory requested. It will also list the contents of any subdirectories if the `recurse` attribute is set to `true`. Unlike the `<cfdirectory>` tag, which acts as a query, `DirectoryList()` creates an array that can be returned much faster.

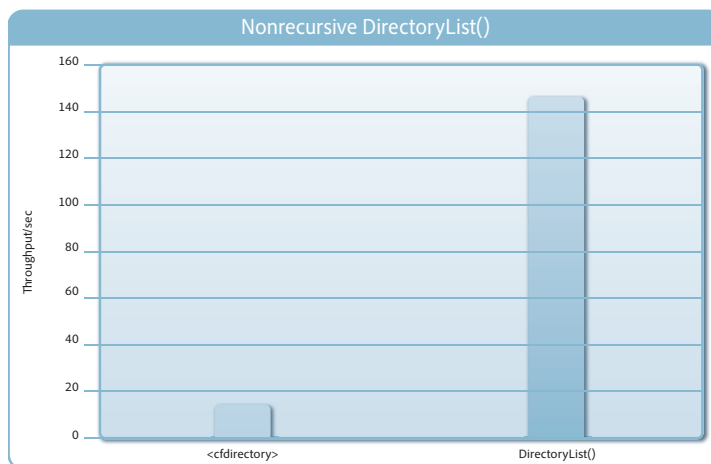


Figure 15. With a nonrecursive, simple listing of 4.5k files, ColdFusion 9 `DirectoryList()` is 10 times faster than the `<cfdirectory>` tag.

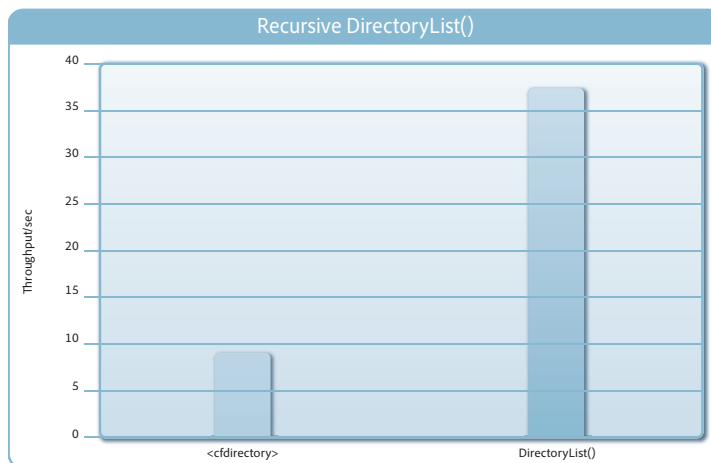


Figure 16. With a recursive listing of 5.5k files in a directory, ColdFusion 9 `DirectoryList()` is 4 times faster than the `<cfdirectory>` tag.

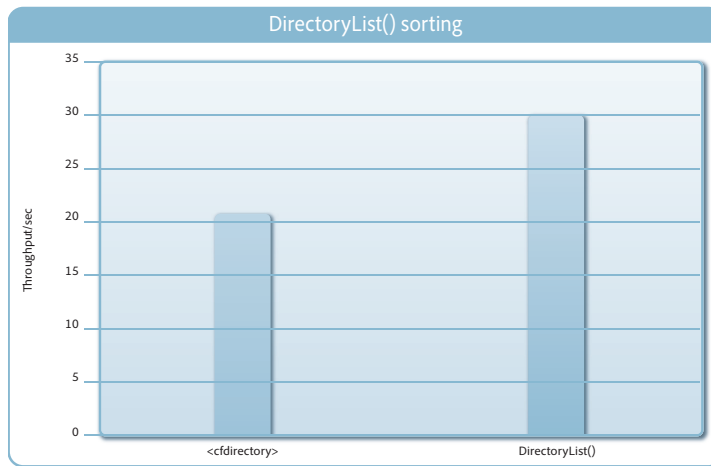


Figure 17. Sorting 600 files with ColdFusion 9 DirectoryList() is approximately 35% faster than sorting by the <cfdirectory> tag.

Getter/setter function—object creation

ColdFusion components support private properties, allowing users to prevent direct access to them. Typically, developers code public getter and setter methods for access to these properties.

In ColdFusion 9, developers no longer have to create getters and setters, as they are created implicitly for every component. Components with implicit getters and setters are created four times faster than components with explicitly coded getters and setters.

Note: Implicit getters and setters can be overwritten if necessary.

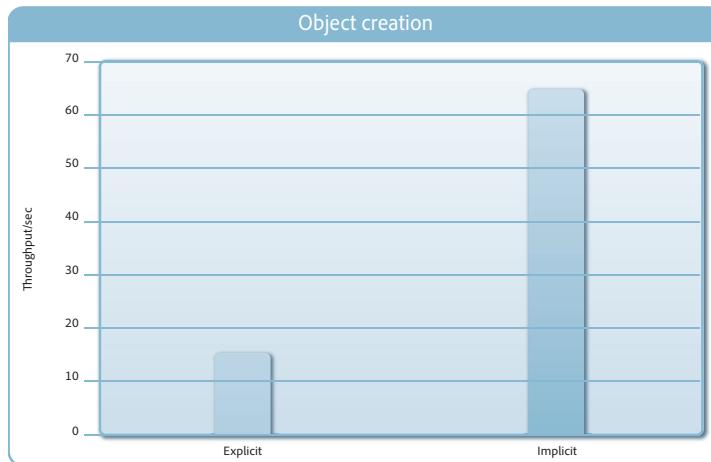


Figure 18. In ColdFusion 9, CFC object creation with implicit getter/setter functions is four times faster than CFC object creation with explicit getter/setter functions.

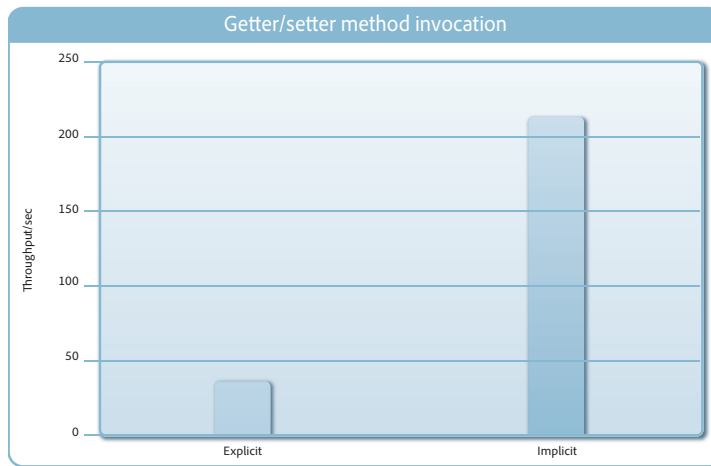


Figure 19. Implicit getter/setter method invocation is seven times faster than explicit getter/setter method invocation.

Enterprise vs. Standard edition

ColdFusion 9 is available in two editions: Enterprise and Standard. Features like spreadsheet and document to PDF conversion use enhanced tuning and a multi-threaded architecture to deliver linear scalability under load and to accommodate high-volume document generation.

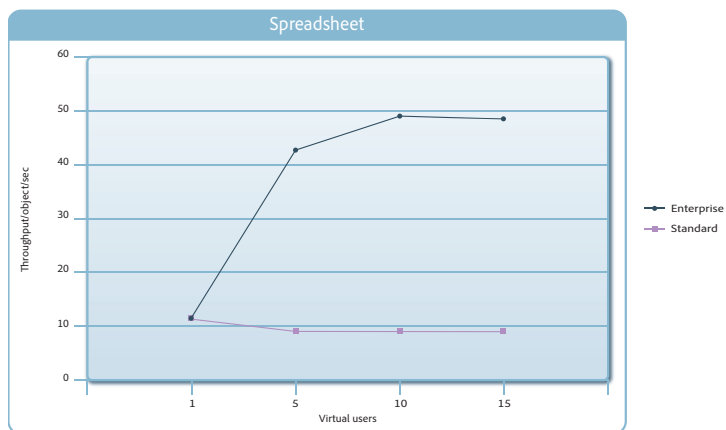


Figure 20. Spreadsheet functionality scales better in the Enterprise edition than in the Standard Edition.

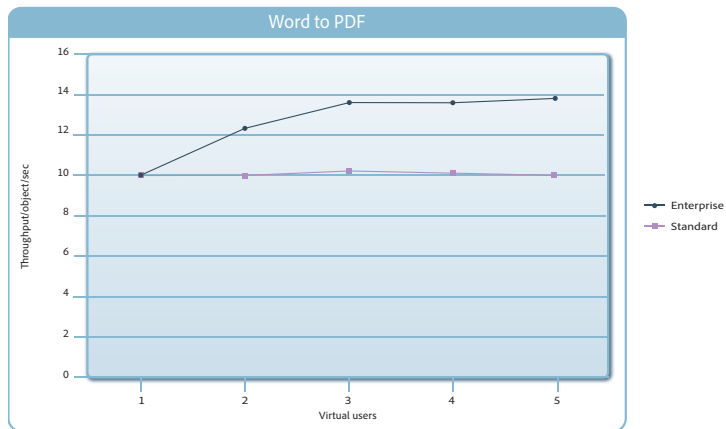


Figure 21. Microsoft Word to PDF conversion scales better in the Enterprise edition than in the Standard Edition.

Summary

Adobe ColdFusion 9 eliminates the bottlenecks in ColdFusion 8, delivering significant performance gains in areas such as CFC object creation/invoke, CreateUUID, evaluate, and Flash Remoting. New features, such as advanced caching, a virtual file system, and a set of directory functions, enable developers to rapidly build, deploy, and maintain higher performing applications.

Appendix: Testing details and additional product information

ColdFusion server settings		
	ColdFusion 9	ColdFusion 8
Java version	1.6.0_14	1.6.0_04
Xmx	1024	1024
Timeout	Disabled	Disabled
Enable white space management	Enabled	Enabled
Disable CFC type check	Disabled	Disabled
Maximum number of simultaneous template requests	30	30
Max running Jrun thread	100	100
Trusted cache	Enabled	Enabled
Cache template in request	Enabled	NA
Component cache	Enabled	Enabled
Debugging	Disabled	Disabled
Server monitoring	Disabled	Disabled

Test configuration

The test applications and the individual performance scripts were run with ColdFusion 8 and ColdFusion 9 on Microsoft Windows 2003 with the following specifications:

- Windows 2003 R2 Enterprise x64 Edition with Service Pack 2
- Microsoft Internet Information Server 6.0
- Four 2.27GHz Intel Xeon® dual-core E5520 processors
- 6GB 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
- 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
- Two 2GHz Intel Xeon dual-core 5130 processors with 1333MHz front-side bus
- 2GB of RAM

Performance measurement tools

Performance measurements were conducted with the freeware Jakarta-jmeter. This tool is used to measure end-to-end performance features like average response time and throughput. For more information, refer to <http://jakarta.apache.org/jmeter/>.

Test methodology

For testing applications, simultaneous HTTP requests were generated to simulate 30 no-think time-virtual users. Each performance test was performed multiple times to confirm the accuracy of the results. The results were averaged and presented in this document.

Migration requirements for Flash Remoting

The following properties should be moved from the ColdFusion "Destination" to "Channel" level. The properties are located under <channel-definition> → <properties> → <coldfusion>. For more details, refer to the Help documentation at http://help.adobe.com/en_US/ColdFusion/9.0/Developing/WS5B9C73A8-5FA2-4a54-B0C6-CECA2E20052D.html.

```
<access>
    <use-mappings>
    <method-access-level>
</access>

<use-accessors>
    <use-implicit-accessors>
</use-accessors>

<use-structs>
    <property-case>
        <force-cfc-lowercase>
        <force-query-lowercase>
        <force-struct-lowercase>
    </property-case>
</use-structs>
```

Note: If XML configuration files (inside WEB-INF/flex) from ColdFusion 8 are dropped without ColdFusion 9 changes, Flash Remoting will continue to work as in ColdFusion 8, without performance gains.

Simple caching

Some ColdFusion pages produce output that changes infrequently. Normally, when ColdFusion receives a request for a page in the application, it executes all the business logic and displays the requested content. If the results change infrequently, it is an inefficient use of processor resources and bandwidth.

The `cfcache` tag tells ColdFusion to cache the HTML that results from processing a page request in a temporary file on the server. This HTML does not need to be generated each time the page is requested. When ColdFusion receives a request for a cached ColdFusion page, it retrieves the pregenerated HTML page without having to process the ColdFusion page.

To cache the page results, place a `cfcache` tag on the ColdFusion page before the code that outputs text.

IIS connector improvements

IIS maintains the worker thread pool size of 256 per CPU. (Some threads are used internally by IIS). When a request arrives, one thread is picked from the IIS worker thread pool and used until the request is completed. When a request arrives for the IIS thread, the request is blocked until ColdFusion serves the CFM page. Meanwhile, if a request comes in for other content types like HTTP or images, IIS uses other threads in the pool to serve them. If all IIS worker threads are being blocked by ColdFusion requests, there will be no threads immediately available for IIS to serve pages of other content types.

When a connector receives a request in ColdFusion 9, it picks the native thread from its thread pool and returns the IIS worker thread with IO Completion (the request will be in IO block state). Once the IIS request is freed from the request, it is immediately available to handle the next request and to serve pages of other content types. Once the request is completed by ColdFusion, the connector thread is returned to its native thread pool.

Users can tune their applications using different native connector thread pool size values. By default, the native connector thread pool size is 25. This thread pool is defined as "maxworkerthread" in file [coldfusion_home]\runtime\lib\wsconfig\1\jrun_iis6_wildcard.ini.

For more information

Product details:
www.adobe.com/go/coldfusion



Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110-2704
USA
www.adobe.com

Adobe, the Adobe logo, ActionScript, ColdFusion, Flash, and Flex are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Intel Xeon is a trademark of Intel Corporation in the U.S. and other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

© 2010 Adobe Systems Incorporated. All rights reserved. Printed in the USA.
91022936 5/10