

# New Caching Features in ColdFusion 9

with  
Aaron West

a.west@me.com  
<http://aaronwest.net>  
<http://twitter.com/awest>



- \*received degree in computer science in 2000
- \*Web developer since 2000
- \*member of Team Macromedia/Adobe Community Experts since 2001
- \*certified Flash/ColdFusion developer since 2003
- \*Nashville ColdFusion User Group Manager
- \*site/blog: [www.aaronwest.net](http://www.aaronwest.net)
- \*twitter: [www.twitter.com/awest](http://www.twitter.com/awest)



- \* what is caching and why should you care?
- \* what caching mechanisms existed before ColdFusion 9?
- \* what new caching mechanisms are in ColdFusion 9?
- \* gain experience with ColdFusion 9 caching through 14 hands-on walkthroughs

Disclaimer: this is an intro level session... we'll do more than scratch the surface, but there's a lot of caching features in CF9.



From Wikipedia:

“a collection of data duplicating original values stored elsewhere or computed earlier, where the original data is expensive to fetch (owing to longer access time) or to compute, compared to the cost of reading the cache. In other words, a cache is a temporary storage area where frequently accessed data can be stored for rapid access. ”

...in other words, a cache is a temporary storage area where frequently accessed data can be stored for rapid access



- \*two things to keep in mind
  - \*cached data is duplicate data
  - \*the duplicate data is stored in a lesser expensive place to be retrieved



- \*data and/or data access at the database level
- \*application level (ColdFusion) data, queries, and/or Web pages
- \*full HTML pages on your Web Server
- \*full HTML pages, images, video on a caching server
- \*just about anything in the cloud (Amazon, Akamai, Limelight)
- \*client-level caching: cookies, rendered Web pages, data



- \* when something...
- \* is frequently accessed
- \* doesn't change often
- \* takes a long time
- \* is universal



- \* cache as close to the final state as possible
- \* cache full pages whenever feasible and possible
- \* cache static files on your Web server, media server, or CDN
- \* use content generation where appropriate
- \* keep four eyes on your cache size





- \* **cache hit** - cache is checked for desired datum. If datum exists it is retrieved
- \* **cache miss** - cache is checked for desired datum but does not exist. Datum is retrieved from main data store such as a database
- \* **hit rate** - how often a searched-for datum is found in cache
- \* **cache latency** - how quickly data is returned from cache when there's a cache hit
- \* **vertical scaling** - adding resources to a single node such as RAM, disk space, or CPUs
- \* **horizontal scaling** - adding additional physical or virtual machines



- \*store things in Application, Session, Server, or Client scopes
- \*cache simple datums such as strings and numbers
- \*cache complex datums such as structures, queries, CFCs
- \*very powerful and easy to maintain

```
<cfset this.applicationtimeout = CreateTimespan(1,0,0,0)>  
.  
.  
.  
<cfset Application.myQuery = GetEmployees>
```



- \* native to ColdFusion (<cfquery>)
- \* very easy to implement with significant performance gains
- \* no visibility into the cache
- \* cache misses result in immediate re-cache which can create problems (dog pile!)
- \* cache store is server wide
- \* <cfobjectcache action="clear">

```
<cfquery name="getEmployees" cachedwithin="#CreateTimeSpan(0,1,0,0)#">  
    [query-here; different query = different cache]  
</cfquery>
```



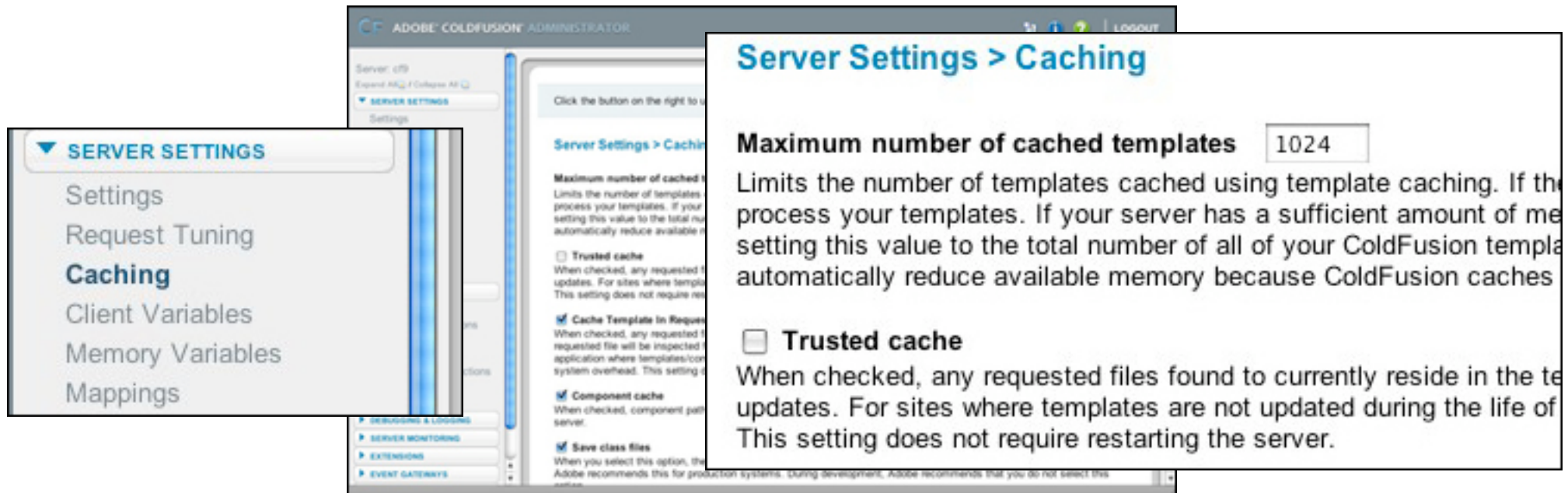
- \*available in ColdFusion 8 and prior
- \*typically used `<cfsavecontent>` and a shared scope such as application or session
- \*also accomplished with custom tags such as scopecache
- \*useful in caching one block of heavy code for all users
- \*requires programmatic cache management



- \*admirable caching strategy but often unrealistic
- \*uses <cfcache> which was added in ColdFusion MX (version 6)
- \*prior to ColdFusion 9 <cfcache>...
  - \*stored cache on disk, the most expensive medium
  - \*lacked granular cache control



- \*max number of cached templates
- \*trusted cache
- \*component cache
- \*save class files



- \* CacheBox by Isaac Dealey
- \* Cache Management by Cristian Costantini
- \* FuseCache by Matt Gersting
- \* all of these are active projects on riaforge ([riaforge.com](http://riaforge.com))



## \*in-process caching

- \*cache operates in and is stored in the same process as app
- \*super fast
- \*considerations
  - \*RAM is a real limitation; monitor closely!
  - \*cannot vertically scale by adding RAM

## \*distributed caching

- \*cache operates in its own process
- \*can scale horizontally and vertically
- \*considerations
  - \*slower due to serialization/deserialization of data (SOAP/REST)

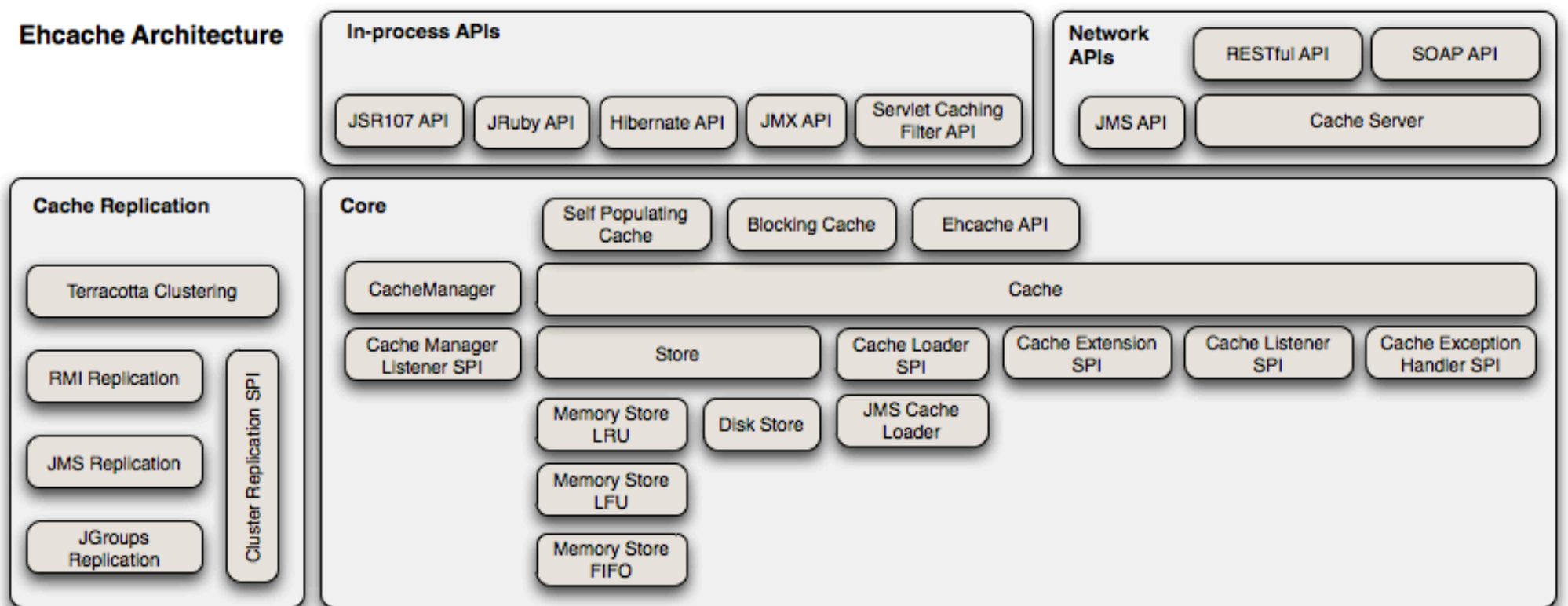




- \* aka replacement policies
- \* every algorithm has some level of compromise between cache hit rate and cache latency
- \* Belady's algorithm (clairvoyant algorithm)
- \* LRU (Least Recently Used)
- \* MRU (Most Recently Used)
- \* FIFO (First-in, First-Out)
- \* LFU (Less Frequently Used; unique to Ehcache)



## Ehcache Architecture



- \* java caching engine for general purpose caching
- \* pronounced “ee h kay sh”
- \* open source; governed by the Apache 2.0 license
- \* an in-process cache; limited by RAM (JVM)
- \* also available as a standalone WAR (Web Application Archive)
  - \* accessed via SOAP or RESTful services
- \* provides LRU, FIFO, and LFU (the special guy)
- \* cached items stored as key / value pairs



- \* can failover cache to disk when RAM fills up
- \* supports cache replication for clustered setups via RMI (Remote Method Invocation), JMS (Java Messaging Service), JGroups, or Terracotta
- \* can be implemented as a distributed cache\*
- \* Ehcache is used by
  - \* Adobe ColdFusion 9
  - \* Magnolia - a leading content management system
  - \* Hibernate - java object relational mapping (ORM)
  - \* Spring - java inversion of control (IoC)



- \* runs as an embedded server in ColdFusion 9
- \* caches are associated with named applications (Application.cfc or Application.cfm)
- \* caches are not directly associated with scopes
- \* a cache created outside of an application is available to all unnamed applications
- \* a default named cache is defined in ehcache.xml
- \* you can define additional named caches



- \* memcache - <http://danga.com:80/memcached/>
- \* Java Caching System (JCS) - <http://jakarta.apache.org/jcs/>
- \* OSCache - <http://www.opensymphony.com/oscache/>
- \* JBoss Cache - <http://labs.jboss.com/jbosscache/>
- \* ShiftOne - <http://jocache.sourceforge.net/>



- \* Object cache

- \* cache just about anything

- \* nice, granular control over the cache

- \* implemented via `<cfcache>` or functions (`<-----` much better)

- \* Template cache

- \* cache full pages or page fragments

- \* little visibility into the cache

- \* implemented via `<cfcache>` only

- \* ORM (Object Relational Mapping) caching (not covered today)



- \* can cache pages on the server, client, or both (server is default)
- \* different URLs are considered different pages thus different cached item
- \* can retrieve cached pages from a Web server
- \* cache is cleared if you change the code on the page
- \* can cache full pages or page fragments
- \* supports both Object caching and Template caching





## \* <cfcache> attributes

- \* action = [cache, clientCache, flush, get, put, **serverCache**]
- \* dependsOn - comma-separated list of variables
- \* directory - absolute path of cache directory
- \* expireURL - removes a specific URL from Template cache
- \* id - identifier for the cached item
- \* idleTime - decimal number of days or timespan
- \* metadata - retrieves cache metadata
- \* name - variable for a retrieved cache item
- \* timespan - decimal number of days or timespan
- \* value - designates the object to put into cache



- \* cache keys are per application; two apps can use same key
- \* cachePut()
- \* cacheGet()
- \* cacheRemove()
- \* cacheGetAllIDs()
- \* cacheGetMetadata()
- \* cacheGetProperties()
- \* cacheSetProperties()
- \* getAllTemplateCacheIDs() (undocumented)





coding time

- \* what is caching and why should you care?
- \* what caching mechanisms existed before ColdFusion 9?
- \* what new caching mechanisms are in ColdFusion 9?
- \* gain experience with ColdFusion 9 caching through 14 hands-on walkthroughs



# Thank you!

e-mail: [a.west@me.com](mailto:a.west@me.com)

site/blog: [www.aaronwest.net](http://www.aaronwest.net)

twitter: [www.twitter.com/awest](http://www.twitter.com/awest)

