Introduction to Caching

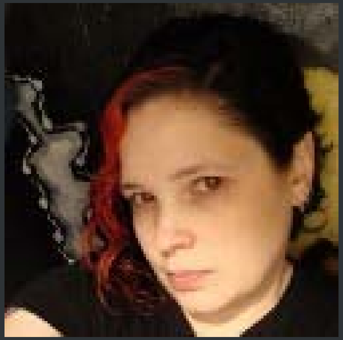
Daria Norris

NCDevCon

September 18, 2011

Who Am I?

- 15 yrs ColdFusion
- 12 yrs as an independent contractor
- Currently work for Free Library of Philadelphia
- CodeBass Radio



Why Do We Cache?

Performance

Scalability

Caching Tools in ColdFusion

- Persistent scopes
- <cfquery> with cachedwithin or cachedafter
- <cfcache>
- Cache within an object (CFC)
- CF9's enhanced caching functions

Persistent Scopes

- Session
- Application
- Client
- Server

Pros

- Easy to use and manage
- Great for simple values

Cons

- Usually needs manipulation
- Can't store ORM Objects

Query Caching

<cfquery>

- cachedAfter=
- cachedWithin=

Pros

- Simple to use
- Performance gain

Cons

- Difficult to manage
- All variants of a query are cached
- Earlier CF version limitations

Page Caching

<cfcache>

- timespan=
- Caches entire page
- Cache managed by URL variables
- Enhanced for CF9

Pros

- Best Performance Gain

Cons

- Won't work for custom content
- Difficult to manage

Fragment Caching

- cachePut()
- cacheGet()
- cacheRemove()
- cacheGetAllIds()
- cacheGetMetadata()
- cacheGetProperties()
- cacheSetProperties()

Pros

- Cache page sections
- Use multiple cached sections on a page
- Easy with CF9

Cons

- Still requires some management
- Names can collide

Object Caching

- CFCs created as objects
- Usually cached into a persistent scope
- Typically called Services

Pros

- Encapsulates complex data

Cons

- Requires cache management
- Possible Race Conditions

Finding the Optimal Caching Plan

- Cache as close to final output as possible
- Cache to static files when possible
- Monitor your cache size
- Play with different techniques until you find the right one

Resources

Ray Camden

http://www.coldfusionjedi.com/

Rob Brooks-Bilson

http://www.brooks-bilson.com/

Adobe

http://help.adobe.com/

Daria Norris @cfGothChic
cfgothchic@gmail.com