ColdFusion Performance Testing and Tuning

By Charlie Arehart
CTO/Founder SysteManage
Our Practice Makes You Perfect

Session Overview

• Introduction to Load Testing Web Applications
• Creating a Real-World ColdFusion Test
• Identifying Common Performance Bottlenecks and Tuning Opportunities
  • CF, Database, Admin, Architecture, and more
• Load Testing Best Practices
• Resources for Learning More
• Questions & Answers

Audience

• Who should be interested in Load Testing?
  – Know you need to test
  – Know you ought to test
  – Don’t think you need to bother testing
  – Have never even considered testing
• In other words, all web application developers
  – You may have reasons to have dismissed it
  – Many common challenges have been overcome

Why Should I Care?

• Even if your application doesn’t expect high load
  • Do you know how it will perform under light load?
  • Do you know how some programming change you’ve made will perform under current production load?
  • Have you wondered what the impact would be of some change in programming or configuration?

The Cold Facts

• Most developers are familiar with the “concept” of load testing, but very few actually do it!
• Most common reasons for not testing are Time, Complexity & Cost (can be very expensive)
• Most sites run into performance problems with only a few users (you don’t have to be ToysRUs)
• Most performance problems are discovered too late in the game to properly address the issues

What does it mean for your site to perform or scale well?

• Delivers page views to users in 3-8 seconds or less (varying industry standards)
• Performs consistently throughout a visitor’s session
• Response times scale linearly as user load increases as opposed to exponentially
What is a Load Test?

- Simulates multiple users to measure, define, validate and maintain optimal application performance, scalability and reliability
- Serves as a compass for understanding the limits of any Web-enabled application and, subsequently, for managing its performance and growth

Virtual User Simulation

- You don’t need to gather users together to do a test!
- Load testing tools can create “virtual users”

Create a Real-World Example in ColdFusion

Customer Case Study:
- Set up a User Profile
- Configure a Load Test Scenario
- Schedule a Load Test
- Analyze Test Results

Load Testing over the LAN vs. the Internet

LAN Testing:
- Eliminates network variability due to the Internet
- Saturates system rather than the network

Internet Testing:
- Uncovers hidden delays due to Internet “weather”
- Exercises your pipe, routers, firewalls and load balancers in addition to the system under test
- Ensures Service Providers are meeting SLAs

Step 1: Recorder

- Setup a “User Profile”
  - This describes how a user might browse through your site
  - Literally, like turning on a VCR, you’ll record your steps through the site
- Record/stop/pause/resume
- No scripting required!
Step 2: Controller

- Configure a Load Test “Scenario”
  - This describes the characteristics of the virtual users you want to simulate
    - Number of users/duration of test
    - Browser type/connection speed to simulate
    - Varying form input data
    - And more
Step 3: Scheduler

- Schedule a Load Test
- Run a test immediately
- Schedule one for a future time
- And more

Step 4: Analysis

- Analyze Test Results
- View the performance of your web app
- Provides multiple graphs for review
- Each graph offers variations of how/what data is presented
  - Can graph multiple types of data at once
  - Can even graph multiple test runs at once
Additional OpenLoad Features

- Supports Cookies and sessions
- Supports SSL processing

Planned Enhancements:
- Monitoring and reporting performance stats
- Providing ongoing monitoring
- Much more

Key Benefit for CF Developers

- Easy to use
- No software installation required
- Puts testing into developers hands
  - No longer the province of “the test team”
  - Allows you to perform testing throughout application life-cycle
- Load test from development to deployment

Why Load Test from Development to Deployment?

- Identifies problems early on before they become costly to resolve
- Reduces development cycles
- Produces better quality, more scalable code
- Prevents revenue and credibility loss due to poor Web site performance
- Increases customer satisfaction and retention
- Enables intelligent planning for future expansion
Maybe High “Load” Isn’t a Problem You Foresee

- Can load testing still be useful for you?
- Can certainly look to common performance tuning opportunities as sources of bottlenecks in high or even low load situations

Common Tuning Opps

- Determine impact of various opportunities:
  - CF coding changes
  - Database design/config/coding changes
  - CF version upgrade
  - CF Admin configuration changes
  - Web server configuration changes
  - System architecture changes
  - HTML coding changes

Common Tuning Opps: “Classic” CF Coding Issues

- Certain classic CF tuning questions may not be that important compared to many others we will discuss
  - Using CFOUTPUT around each var or not
  - Cost of pound signs over-use
  - Prefixing all variables
  - Using CFSCRIPT where possible
  - Cost of Evaluate() vs IIF() functions

Common Tuning Opps: Other CF Coding Issues

- Consider benefits of caching pages (or parts of page) that are rather static: CFCACHE, CFSAVECONTENT
  - Be careful about using .cfm just to include static navbar
  - Consider page caching instead:
    - Consider impact of locking (and not locking) all persistent variable use. Big debate. Load test to know.
    - Be careful with setting timeouts for CFLOCK, CFHTTP, CFQUERY too high

Common Tuning Opps: More CF Coding Issues

- Alternatives for reading text files: CFFILE vs CFHTTP vs ODBC text drivers vs Merant text drivers vs Java custom tag
  - Don’t enable clientmanagement="yes" if not using client variables
    - Writes client vars about every visit to registry (or DB)
  - Consider cost of CFINCLUDE vs CF_c- custom tag vs CFMODULE vs CF vs CFS object vs UDF vs CFM’s CFCs and CFFUNCTION options
  - Be careful of recursion in UDFs, custom tags

Common Tuning Opps: DB Design/Config

- Database processing is usually largest contributor to poor performance
  - Starts with choice of right DBMS for the job
  - Proceeds to good database design
  - Proper use of indexing can have big perf impact
  - Use DBMS options to test/tune your SQL and DB design
  - Also configuration of datasources
    - Consider cost/benefit of “maintain db connection”
    - Tune “Limit simultaneous access to db to x”
    - Consider ODBC vs OLEDB vs Merant vs Native driver
    - Tune Admin’s “limit cached db connection inactive time to x minutes” (global setting)
Common Tuning Opps: DB Coding
- Biggest goal is to avoid unnecessary DB I/O
- Also, don’t do in CF what can be done better in DBMS/SQL
- Avoid SELECT *; use count(*) vs recordcount
- Use aggregate functions, inner and outer joins; unions
- Consider subqueries (in SELECT and WHERE)
- See Ben Forta’s “Teach Yourself SQL in 10 Minutes”
- Use CFOUTPUT GROUP where appropriate
- See CFML docs for more info

Common Tuning Opps: DB Coding (cont.)
- Use DB caching where appropriate
  - CACHEDWITHIN and CACHEDAFTER on CFQUERY
  - Caching to persistent scopes (session/application/server)
  - CF5’s query of queries capability
- Consider stored procedures for faster execution
  - Consider CFQUERY call of SP vs CFSTOREDPROC
- Consider impact of CFTRANSACTION IsolationLevel

Common Tuning Opps: CF Version Upgrade
- Upgrade to CF5
  - Up to 5x faster than CF4.5
  - Reduced memory footprint
  - Better memory management
    - Fewer leaks, better release of unused resources
  - CFMX has just come out
  - Remains to be seen what performance impact will be

Common Tuning Opps: CF Admin Config
- Tune “limit simultaneous requests”
- Tune template cache size, consider “Trusted Cache”
- Tune “maximum cached queries”
- Weigh impact of “auto read locking”, “full checking” of session, app, and server scope variables
- Single-threading of sessions
- Disable “Debugging>Enable PerfMon/Enable StackTrace” options if not using them
- Consider turning on “strict attribute validation”
- Consider impact of Client vars stored in ODBC DB

Common Tuning Opps: Web Config
- Web server limitations (IIS on Win2k Pro supports only 10 users by default, for instance)
  - See Macromedia KB article 20075
- About tuning the IIS “Application Protection” setting
- Don’t use SSL pages where they’re not needed
  - Ok for credit card/privacy info request processing
    - But don’t put whole site or section under SSL if not needed

Common Tuning Opps: System Architecture
- Consider benefits of scaling hardware/software
  - Vertical scaling: adding memory, CPU power, multiple CPUs per box
  - Horizontal scaling: S/W vs H/W load clustering, round robin DNS
- Consider tiered server configuration
  - Separating database server from web/cf server
  - Possibly creating separate image server
  - Possibly sharing a single file/template server for multiple clustered CF servers
**Common TuningOpps:**
**HTML Coding**
- Over design of Web pages (graphics: too many, large)
- Pre-load images before they are needed
- Validate on the client whenever possible (JavaScript or ActionScript)
- Observe design & coding best practices

**Performance Opportunity:**
**Monitoring**
- Things you can monitor in CF
  - Log slow pages (remember 8-second rule)
  - Enable Debugging (to check processing time in code)
- Things you can monitor with O/S tools (PerfMon)
  - Memory contention issues (paging)
  - Heavy disk I/O (high service times)
  - Excessive queue lengths (more than one) & wait times (more than a few milliseconds)
- Other possible monitors
  - Network latency (packet collisions & lost)
- Later releases of OpenLoad will show such monitoring

**ImprovingApparent Response Time**
- Consider suppressing whitespace
  - At admin level, with “SuppressWhiteSpace” option
  - In code, with CFSETTING, CFSILENT, and CFPROCESSINGDIRECTIVE tags
- Use CFFLUSH
  - Can cause part of page to display to user before entire page is completed
- Leverage browser caching where appropriate (CFHEADER)
- Consider gzipping output from CF templates
  - Can set at web server level, can also program
  - May be even easier to do in CFMX with Filters

**More TuningOpportunities**
- These have been just a few of the more common tuning opportunities
- Not all will make sense for all sites/apps
- There are still more
  - Many of which may not be obvious but may be valuable
- Planning to create a benchmark site
  - To be published showing impact of various changes
    - Using a standard application suite
  - Still need to test/tune these things for yourself

**Leverage Knowledge of Others**
- Perform design and code walkthroughs
- Keep up on, apply industry best practices, coding stds
- Participate in user groups
- Read the CF Dev Journal magazine
- Join email mailing lists

**Resources for Learning More**
- Macromedia Resources
  - ColdFusion 5 Performance Brief
  - CF 4.5.1 Performance Tuning Brief
  - Administering ColdFusion Server, Chap 6
    - “Creating Scalable and Highly Available Web Sites”
  - Macromedia Knowledge Base/TechNotes
    - 922 (somewhat dated), 556, 8627 (detailed tuning ideas), 12970 (general load testing guidelines)
    - http://www.macromedia.com/v1/support/knowledgebase/search.cfm
Resources for Learning More

- Optimizing ColdFusion, Chris Cortes, from Osborne/MCGraw-Hill
- Certified ColdFusion Developer Study Guide, Ben Forta
  - Chapters 36, 37 and others
- My Jan 2002 CFDJ article on Admin Performance Settings
- Microsoft Duwamish App Perf Site
- Macromedia KB 11773 points to several resources
  - many outside of Macromedia

Load Testing Do’s

- Do test early and do test often
- Do establish what is and is not acceptable performance for your application
- Do test from the user’s perspective - it’s the only one that counts
- Do baseline and compare your findings
- Do monitor your system while you test
- Do test whenever there’s a change in your site’s content, code or infrastructure

Load Testing Don’ts

- Don’t wait until the last minute to test
- Don’t depend on your customers to do your testing for you
- Don’t test under unrealistic conditions
- Don’t forget that increases in site signups, accumulation of history, table sizes, disk usage and network traffic will degrade your application’s performance over time
- Don’t be so quick to throw hardware at the problems you turn up - it doesn’t always help

Test your app for FREE

- Register at http://www.opendemand.com/cf/
- Test for FREE. Choose either:
  - up to 25 virtual users for one hour
  - or up to 5 virtual users for a currently unlimited time period
- Obtain online copy of this presentation
- Try-out performance tuning tips
  - Test again and compare results to your baseline

Q&A

carehart@systemanage.com
http://www.systemanage.com

info@opendemand.com
http://www.opendemand.com/