

Welcome

To Advance through Presentation
Use Page Up and Page Down Keys



99 | Worldwide
Developers
Conference

WebObjects



“This year’s Software Product of the Year is Apple Computer’s WebObjects 4.0... Along with substantial power and flexibility, WebObjects brings an ease of development rarely seen in industrial-strength development environments.”





Fine-Tuning the Performance of WebObjects Applications

Dave Neumann
Consulting Engineer

Anand Sundaram
RSW Software



Fine-Tuning the Performance of WebObjects Applications

Bruce Arthur

Manager, Enterprise Objects

Mai Nguyen

Developer Support

Agenda

- Measurement
- Performance Tips
- System Performance with RSW
- Q&A



Where Is the Pain?

- I/O and processing between database and application
- WOF Request/Response handling
- Your custom actions
- I/O between browser, web server, and application



Measure First

- System granularity
 - Use WOPlayback and Professional Tools like RSW e-Load
- Page granularity
 - WOStats page
- Sub-page granularity
 - WOAssociationTiming



Measure First

- Memory
 - ObjectAlloc and OptimizeIt
- SQL
 - EOAdaptor Hooks
 - DB Server features





Demo

First Glimpse: RSW, WOStats,
WOAssociationTiming,
ObjectAlloc



99

Worldwide
Developers
Conference

Database I/O

Optimize Your DB I/O (1)

- What are you fetching?
 - Use `EOAdaptorDebugEnabled = YES` to view SQL statements
 - Determine if performance problem is during fetching or saving by studying the SQL
- Maybe you can fetch fewer rows?
 - Set a Fetch limit for each fetch or application wide



Optimize Your DB I/O (2)

- Maybe you can avoid I/O completely?
 - Take advantage of relationships in your EOModel and object uniquing
 - Fetch once, and filter; take advantage of in-memory sorting
 - Share display groups in session instead of per page; reuse filters and sorts
 - Use PK blocks
 - Implement a result-set cache (large DBs but slowly changing data)



Optimize Your DB I/O (2)

- Hooks for result-set caching:
 - Make yourself the EODatabaseContext delegate
 - Implement:
`databaseContextShouldFetchObjects()`
[for using cache instead of DB]
 - And: `databaseContextDidFetchObjects()`
[for setting cache after a DB fetch]
 - Use a description of the `fetchSpec` as your key



Prefetching; Batch Faulting

- User access patterns
 - Faults are efficient (you don't fetch what you don't need; next accesses cached)—[The drill-down pattern]
 - Faults are expensive (DB round trip); use prefetching instead to avoid them—[The reporting pattern]
- If the usage pattern is not clear, use batch faulting





Demo

Impact of Prefetching
Setting Prefetching in
EOModeler

Setting Up Batch Faulting in
E OModeler

Raw Row Fetching

- Use raw rows for extremely large fetches—examples:
 - Fetch a slimmed down version of the row
 - Simply take advantage of the faster fetch
 - Fetch data that need no relationships and no logic
- EOF provides API to easily convert raw rows into Enterprise Objects
[objectFromRawRow in EOUtilities]





99

Worldwide
Developers
Conference

Page Processing

Getting After the Pages...

- Look for expensive processing you can do once instead of over and over
 - Can you put it in a constructor instead of awake?
- Change usage patterns
 - Drill-down instead of all-at-once reporting



Syncing the State Yourself

- Avoid syncing overhead in your subcomponents
- Use `synchronizesVariablesWithBindings` to control push and pulling values
 - `valueForBinding()`
 - `setValueForBinding()`
 - `hasBinding()`



Compile Your Most Complex Pages

- Objective-C/Java is faster than WebScript
- If you do have a scripted page, do as much in the .wod as possible to minimize script execution requirements (you are using more native WOF code this way)
- In Java apps, consider compiling your EO's in ObjC and creating Java wrappers from them



Some Tips

- NSString stringWithFormat: very slow (stringByAppendingString about 10x faster)
- Formatters—reuse them; don't re-create
- Use threads carefully
 - For HTTP File Upload or to generate images on the fly
 - Know your app and its usage pattern before applying threads!





Demo

WOAssociation Timing
for Detailed Page Analysis

Can't Make It Any Faster?

- Punt!
 - Use `WOLongResponsePage` to give the user immediate feedback





Demo

WOLongResponsePage
Example



Footprint

Leaks (1)

- Use the WOStats and System testers like RSW to track your application's image size
- Use OptimizeIt to find Java static references
- Use ObjectAlloc to visually inspect memory consumption
- Avoid cycles across the Java bridge





99 | Worldwide
Developers
Conference

Demo

ObjectAlloc

Leaks (2)

- Leaks that aren't
 - Use `WOWorkerThreadCount` and `WOGarbageCollectionRepeatCount` to loop collection during session deallocation
 - Use `WOGarbageCollectionPeriod` to force GC on every 'n' requests
- In Objective-C implement `dealloc`—use nested autorelease pools





99 | Worldwide
Developers
Conference

System Performance

The Big Picture

- Plumbing
- WOAdaptor flavor
- Web server
- Browser to web server
- Performance viewed from the web browser





Load Testing WebObjects Applications

Anand Sundaram

Director of Product
Management and Founder

RSW Software

Agenda

- Web Testing Challenges
- Goals of Load/Scalability Testing
- Overall Load Test Methodology
- Phases of Load Testing
- About RSW Software
- e-LOAD Demonstration



Challenge of Testing Web Applications

- Web applications can be very complex
 - Multi-tier, multi-technology, e.g., HTML, CORBA, Java, RDBMS...
- Applications must scale to hundreds/thousands of users overnight
- Applications are a moving target

Increased risk and a need to test early and often



Goals of Load/Scalability Testing

- Ensure that application will perform well and degrade gracefully under anticipated traffic/heavy load
 - Determine the “Magic Number” of concurrent users
- Ensure that application will not crash under peak loads
- Understand best ways to scale hardware/software as traffic grows



Overall Load Testing Methodology

1. Record several typical user transactions
2. Create multiple “scenarios” that represent typical and worst-case application usage
3. Ramp up # virtual users with typical think times until “magic number” is determined

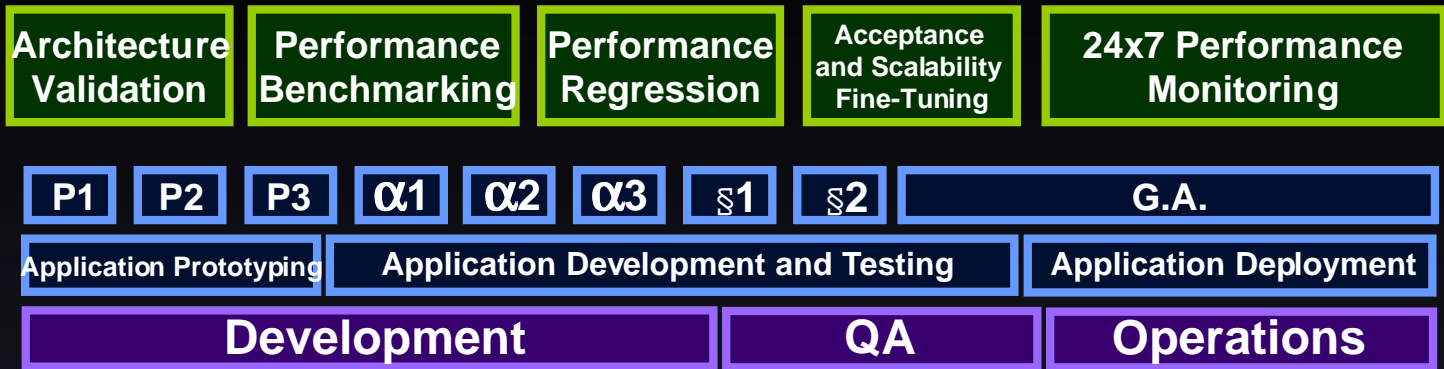


Overall Load Testing Methodology (Cont.)

4. Ramp up virtual users with no delays to determine robustness of application
5. Examine graphs, reports, page times, and system monitors to identify bottlenecks
6. Modify system to remove bottleneck
7. Repeat tests to determine impact of changes to system



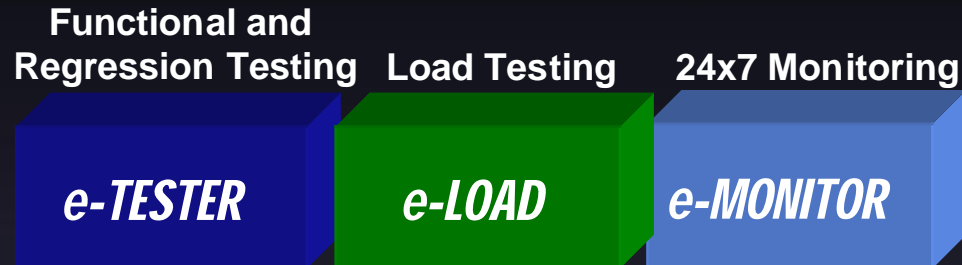
Key Phases of Load Testing



About RSW Software

- Founded in 1996 by 3 ASQ veterans, and became a business unit of Teradyne in November, 1997
- 100% focused on Web application testing solutions
- Products:

e-TEST Suite



WebObjects Sessions

**Complex WebObjects
Application Design**

Hall B
Thur., 9:00am

**WebObjects:
Tips from the Experts**

Hall B
Thur., 10:15am

**Advanced HTML/DHTML/
JavaScript Tips and Tricks**

Hall B
Thur., 4:00pm





Think different.TM



Welcome

To Advance through Presentation
Use Page Up and Page Down Keys



99 | Worldwide
Developers
Conference