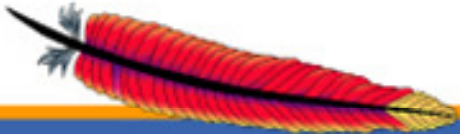


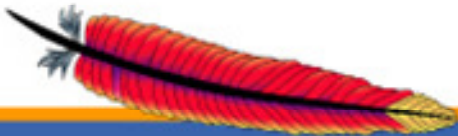
Essentials on Performant JavaServer™ Faces Apps

Client Versus Server Side



What can you expect?

Understand the performance implications of state saving options in JSF



Agenda

1. What is JSF state saving?
2. Methods and options
3. Smack down
4. Facts and Figures
5. Q & A



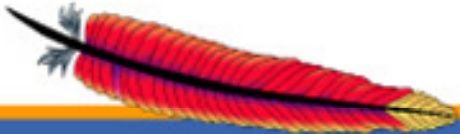
Agenda

1. What is JSF state saving?
2. Methods and options
3. Smack down
4. Facts and Figures
5. Q & A



Web 2.0

- Web 2.0 demands highly interactive apps
- Similar to desktop apps
- Desktop apps are stateful
- For a richer user experience, web apps should be stateful as well

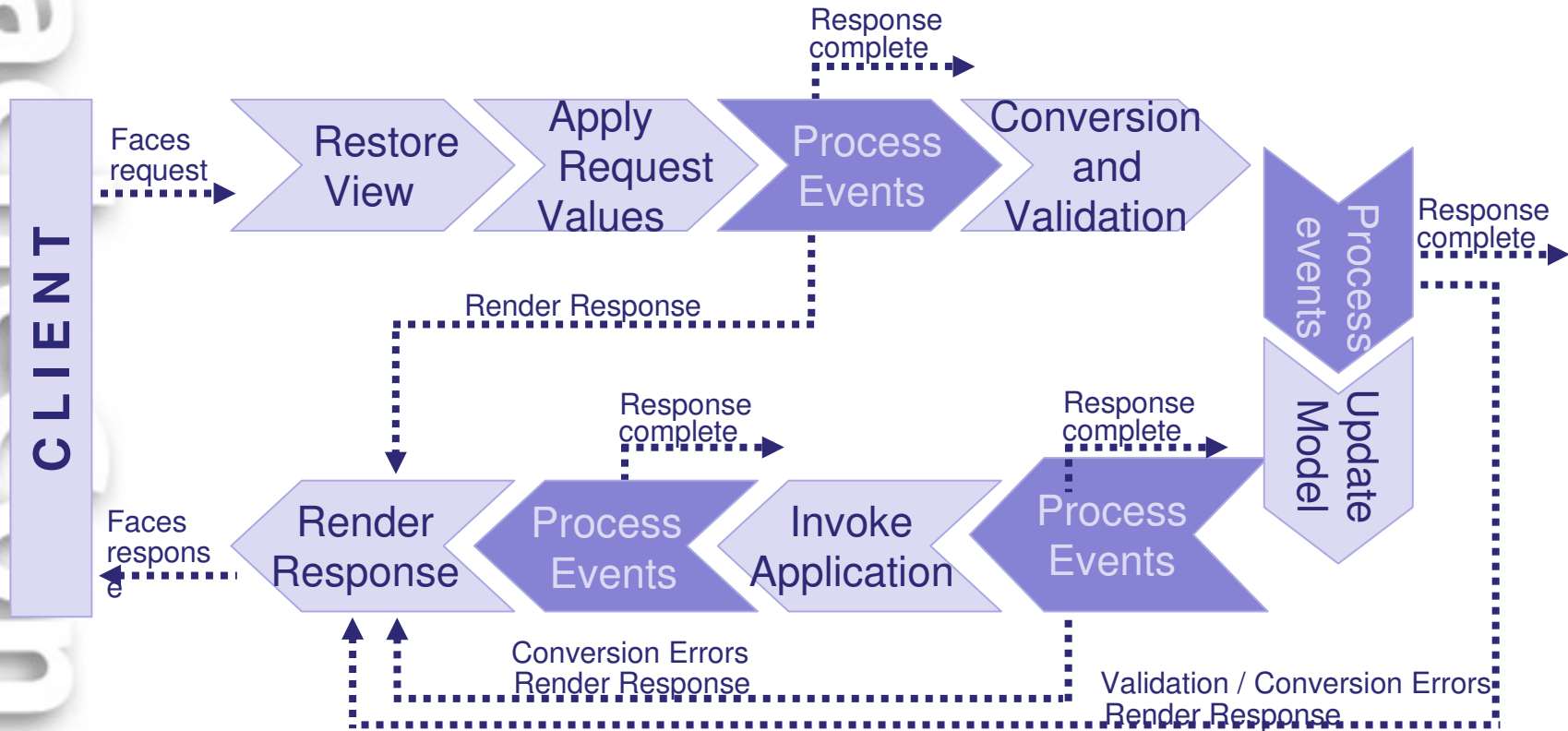


History

- HTTP is the web transport mechanism
- HTTP is stateless
- Web applications are stateless by default

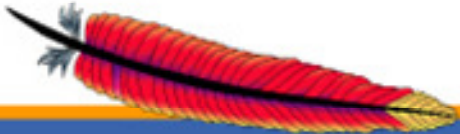


JSF Request Lifecycle



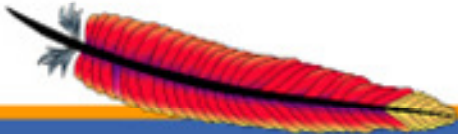
JSF State-Saving Principle

- All view related state in JSF is saved and restored
 - component properties
 - component tree structure
 - part of the domain model (MyFaces option)



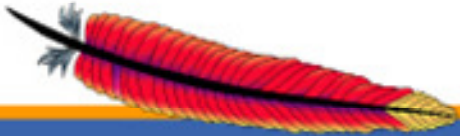
Components take part...

- saveState
 - All non-transient properties saved
- restoreState
 - All non-transient properties restored
- Example
 - Component with properties **title** and **text**



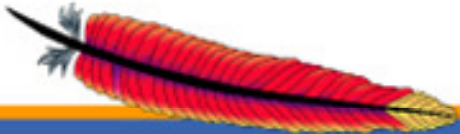
Components - saveState

```
public Object saveState (FacesContext
    context)
{
    Object values[] = new Object[3];
    values[0] = super.saveState(context);
    values[1] = this.title;
    values[2] = this.text;
    return ((Object) (values));
}
```



Components - restoreState

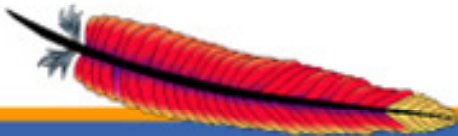
```
public void restoreState (  
FacesContext context, Object state)  
{  
    Object values[] = (Object[])state;  
    super.restoreState(context, values[0]);  
    this.title = (String)values[1];  
    this.text = (String)values[2];  
}
```



State Saving and Performance

Questions *you* should be asking

- CPU performance
- Memory consumption on the server
- Memory consumption on the client
- Bandwidth



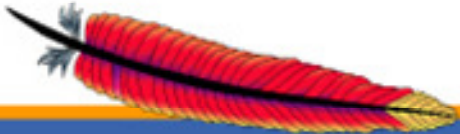
Agenda

1. What is JSF state saving?
2. **Methods and options**
3. Smack down
4. Facts and Figures
5. Q & A



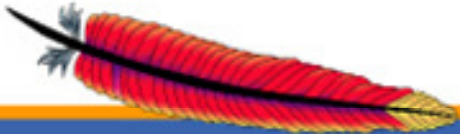
Methods of State Saving

- JSF statefulness
 - server side – servlet session
 - client side – hidden HTML input
- Neward's example
 - server side state saving is context incomplete
 - client side state saving is context complete



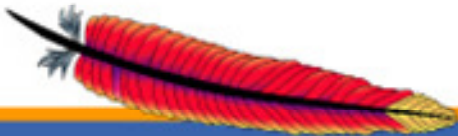
Options in State Saving

- It might be necessary to do:
 - serialization
 - compression
 - encryption
 - encoding
- HTTPS and Client side encryption
- MyFaces offers options to control these



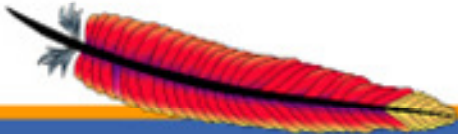
Client Side State Saving

- State is saved into the HTTP response
- Hidden form field when using HTML



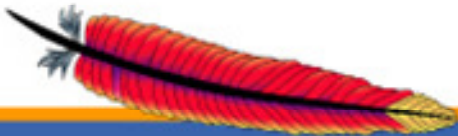
Server Side State Saving

- State is saved into the servlet session
- Options to control the “How”
 - Serialization
 - Compression



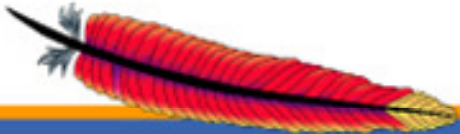
Agenda

1. What is JSF state saving?
2. Methods and options
3. Smack down
4. Facts and Figures
5. Q & A



Server Side vs Client Side

- Server Memory
- Client Memory
- CPU usage
- Browser back button and multiple views
- Bandwidth
- Server restart friendliness
- Security



Advantages of Client Side State-Saving

- No need for Servlet Sessions at all
- No memory consumption between requests
- “F5 browser testing”
- No browser back problem
- No concurrency issues
- Cluster friendlier
- Restart availability



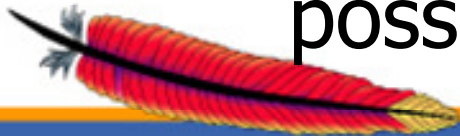
Disadvantages of Server Side State-Saving

- Memory consumption on the server per user in a session
- Memory consumption of several views to support the back-button
- Clustering – moving the session to another node of the cluster
- Development: context is not stored – you need to put the server into the same context again



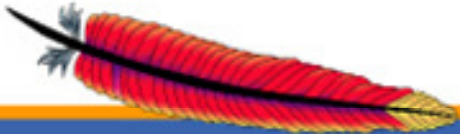
Advantages of Server Side State-Saving

- Bandwidth usage is low
- CPU processing time usage is low
 - Only if you do no serialization
- Smaller problems with AJAX
 - Sending state to server is not necessary
- Application without JavaScript is possible



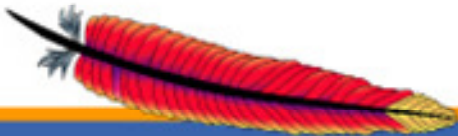
Disadvantages of Client Side State-Saving

- Higher bandwidth
- Higher CPU
- Higher client side memory usage
- Security



Agenda

1. What is JSF state saving?
2. Methods and options
3. Smack down
4. Facts and Figures
5. Q & A



Performance Testing - MyPizza

The screenshot displays the MyPizza website interface. At the top, the site title "MyPizza" is visible. Below it is a navigation menu with links for "Home", "Admin Users", and "Pizza Admin". The main content area is divided into several sections:

- Logged In:** Shows the current user as "Administrator MyPizza" with a "Logout" button.
- Newsletter:** A section for subscribing to the newsletter, including a "Subscribe now" link and an "E-mail:" input field with a "Subscribe" button.
- Contact:** Provides contact information for MyPizza Inc., including the address "Street 3-5", "City", and "Tel. +123124235".
- Create new user:** A form for creating a new user with the following fields:
 - User: admin
 - Password: admin
 - First Name: Major
 - Last Name: Geek
 - Address: MyPizza Street
 - Email: admin@mypizza.com
 - Phone No.: +4912234324325An "Update" button is located at the bottom of the form.

At the bottom of the page, there is a copyright notice "Copyright (c) 2005" and the site name "MyPizza".



Performance testing - Configuration

- **Software:**

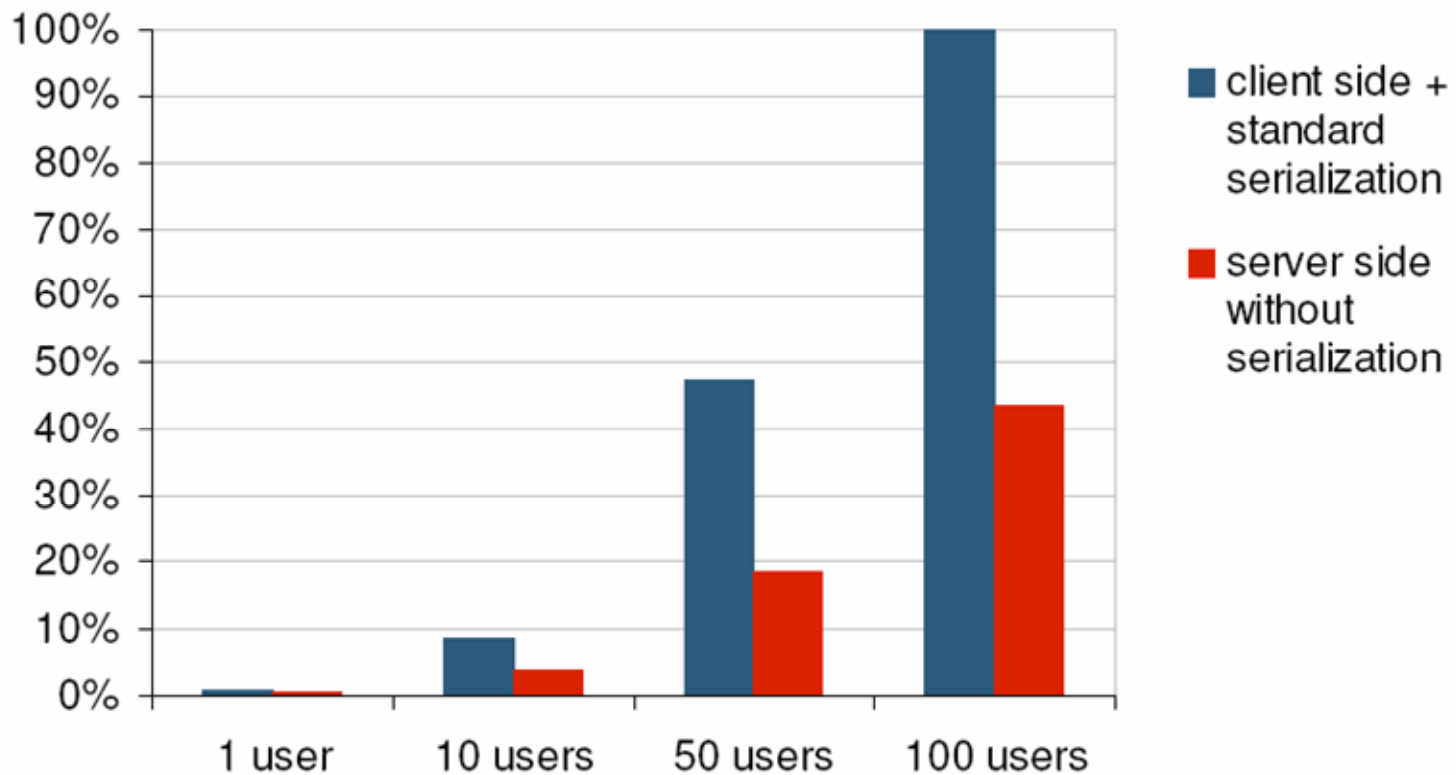
- JMeter 2.1.1 running on J2SDK 1.4.2_03-b2
- with default jvm settings
- Tomcat 5.0.28 running on JDK 1.5.0_04
- with jvm settings -Xmn128M -Xms256M -Xmx512M (all rest default)
- MyFaces 1.1.3

- **Machines:**

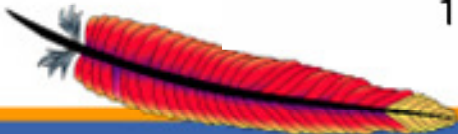
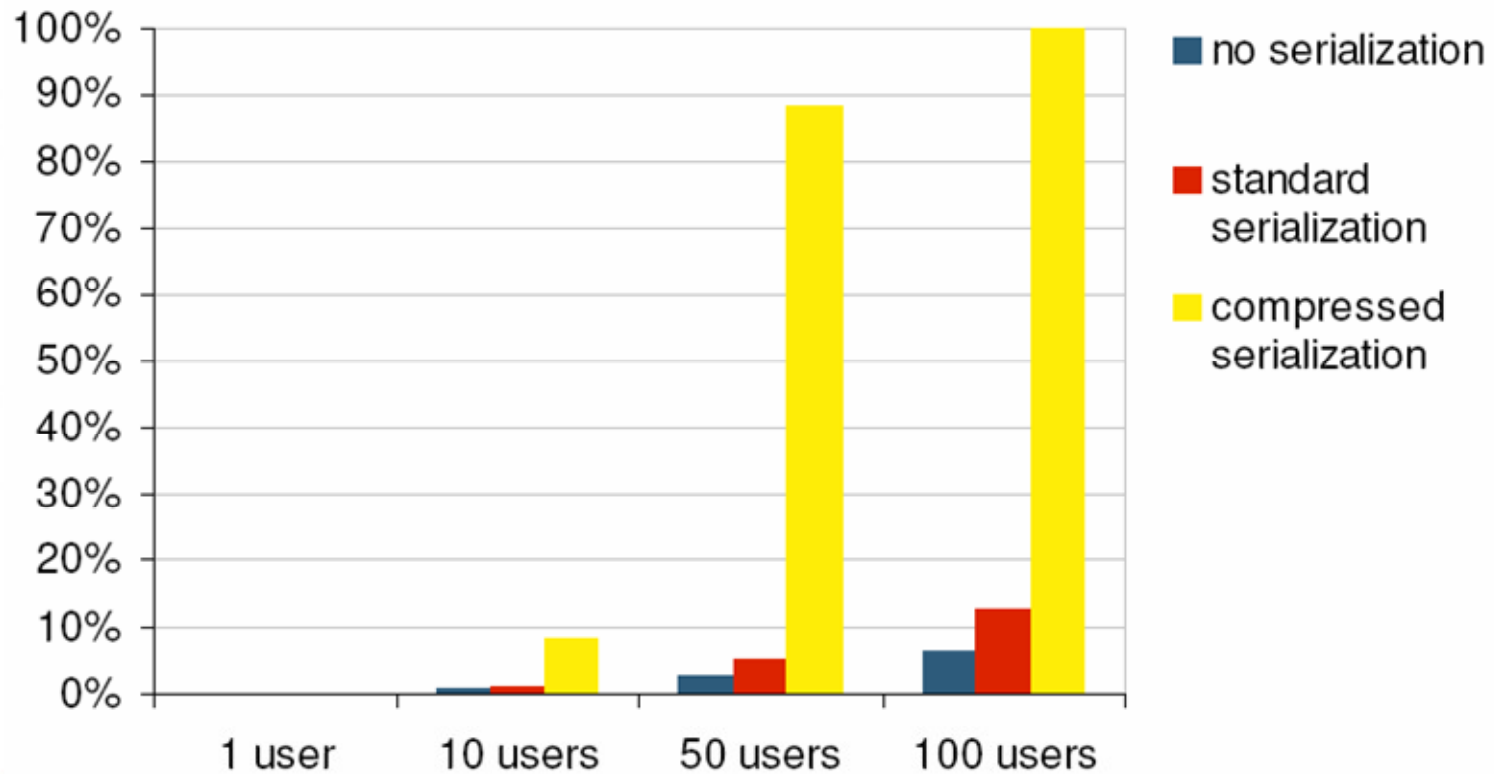
- JMeter running GUI mode on AMD Athlon 1.6GHz, 1GB DDR
- Tomcat running on Pentium 4, 2.79GHz, Hyperthreading, 1GB DDR
- (Dell Dimension 5000)
- LAN 100Mb/s



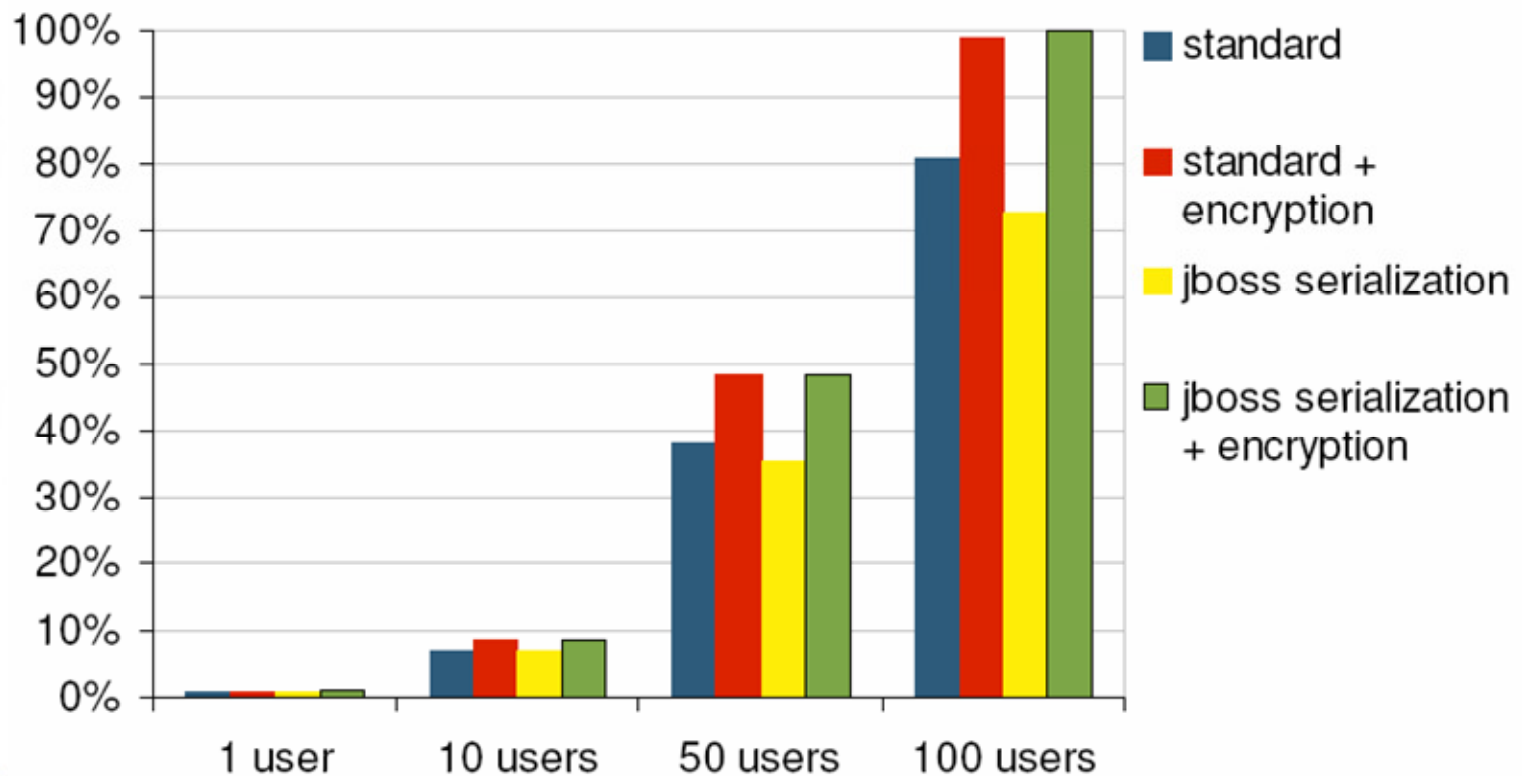
Client side vs. Server side Benchmark Results



Server Side Benchmark Results



Client Side Benchmark Results



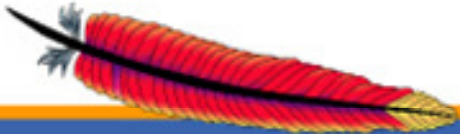
Summary

	Server Side	Client Side
CPU Usage	✓	
Server Memory		✓
Client Memory	✓	
Bandwidth	✓	
Security	✓	
Back Button		✓



Additional Performance Tweaks

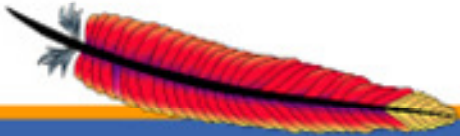
- Streaming Resources option
- Facelets
- JBoss Serialization



ApacheCon

Questions?

Answers!!!



US 2006

Finish

Thank you for your
Attention!

