Navigating the Banner XE Technology Stack

Brian Schoudel, Dir Application Administration
John Tanner, Enterprise App Admin Specialist
University of Illinois
April 19, 2016
Session 36090
Session rules of etiquette

• Please turn off your cell phone/pager

• If you must leave the session early, please do so as discreetly as possible

• Please avoid side conversation during the session

Thank you for your cooperation!
Agenda

1. Past/Present/Future with Banner XE
2. Banner XE architecture
3. Banner XE modification process
4. Tips for Troubleshooting XE
5. Summary
Past/Present/Future using XE
University of Illinois

- Three campuses (Urbana-Champaign, Chicago, Springfield) [www.uillinois.edu](http://www.uillinois.edu)
  - Urbana-Champaign – 44K students
  - Chicago – 28K students
  - Springfield – 5 K students
- Production Banner database 1.6 TB
- 5.8 million spriden recs
Current Banner Prod Environment

- Been live since 2001
- Licensed for everything but Alumni
- Banner 8 releases kept up to date (Gen 8.8 this month)
- OFM 11g for app servers on RHEL 6
- Oracle 11.2.0.4 Sun M9000 for database server
- Multi-Campus VPD for Fin Aid, Academic Hist, Web Tailor, Course Catalog and Schedule
- FGAC used throughout Student and HR to segregate by campus, college, dept, access, etc..
Current XE prod environment

- Database Extension Util (DBEU)
- Student Faculty Grade Entry 9.3 (Active)
- Student Course Catalog 9.3
- Student Class Schedule 9.3
- Student Advisor Self Service 9.1
- Student Advisor 9.1
- Student API 9.1.1
- Student Overall 9.3
- Student Registration 9.2 (Limited)
- Student eTranscript 9.0
- General Events Mgmt 9.3 (Active)
- Banner DB Upgrade 9.1
Banner XE timeline

- (6/2012) DBEU
- (9/2014) Fac Grade Entry 9.1
- (9/2014) Catalog/Schedule 9.0
- (11/2014) eTranscripts 9.0
- (5/2015) Advisor 9.0
- (5/2015) Events 9.3
- (10/2015) Registration 9.2
- (10/2015) Advisor 9.1
- (10/2015) Student API 9.1.1
- (10/2015) Catalog/Schedule 9.3
- (10/2015) Student Overall 9.3
- (4/2016) Registration 9.3
- (4/2016) Fac Grade Entry 9.4
- (4/2016) Student Overall 9.4
- (4/2016) Student API 9.4
- (4/2016) Academic History 9.1
- (4/2016) Catalog/Schedule 9.4
- (4/2016) Student SSB 9.2
Other apps in test/beta…

- Ellucian Solution Manager 1.5
- Application Navigator 1.3.1
- Banner Finance Transformed Forms 9.0 (Beta)
- Banner HR/Payroll Transformed Forms 9.0
- BEIS SSO Manager
- Ellucian Identity Services (SAML authentication)
- Ellucian Integration Hub (Beta)
Miscellaneous Notes on XE

• Need to stay current with Banner 8
• It is modular but beware of dependencies
• You can implement only the apps you want (can also continue to use Banner 8 and ease into it)
• Been slower to adopt then we initially thought.
• Really version 1.0, 1.1, … (not 9.0, 9.1, …) of a new application (should get better over time)
Banner XE Architecture
Banner XE Architecture

• Why WebLogic?
  
  o Oracle Site License
  
  o 24X7 Oracle Support
  
  o WebLogic Familiarity (Existing Forms/Self-Service Environment)
Banner XE Architecture

• Environment Setup
  
  o Four Red Hat Enterprise 6.7 Linux Servers
    • Duplicate Hardware setup in Development, QA, and Production
    • 64 GB RAM
    • 16 CPU’s

  o WebLogic Server 10.3.6.0.13

  o Oracle HTTP Server 11.1.1.9.0 (2 OHS instances per server)

  o Server Iron Hardware Load Balancer
Banner XE Architecture

• Web Tier Setup

o OHS1 used for all non-administrative applications. No firewall restrictions.

o OHS2 used for all administrative applications. Firewalled to campus users.

o Multiple Development/Test environments running simultaneously.

o Banner instances differentiated with -SIDNAME

o Web Tier mod_wl_ohs.conf used to route application to specific WebLogic managed servers.
Banner XE Architecture

- Managing Banner databases
  
  - Modification configuration files to maintain –SIDNAME
  
  - XE
    - instance.properties (shared_configuration directory)
    - XEAPP_configuration.groovy (application logging)
    - weblogic.xml (extracted from war)
    - web.xml (extracted from war or manually edit CreateWar.groovy to update self-service datasource.)
Banner XE Architecture

• Managing Banner databases (continued)
  
  o Modification configuration files to maintain –SIDNAME
  
  o WebLogic
    
    ▪ mod_wl_ohs.conf
      
      <Location /BannerGeneralEventManagement-BANDEV>
        SetHandler weblogic-handler
        WebLogicCluster managedserver01:8088,managedserver02:8089
      </Location>
    
    ▪ ssl.conf
      
      RewriteRule ^/BannerEventManagement-BANDEV/login/auth/* https://authentication-provider [R]
Banner XE Architecture

• Issues managing multiple Banner databases
  o Configuration file management.
    • Creation of scripts to centrally manage and maintain mod_wl_ohs.conf and Banner instances between application servers.
  o “Banner Already Open in Another Window” error.
    • Manual extraction and editing if WEB-INF/weblogic.xml cookie-path.
  o Managed server application deployment tracking.
    • WebLogic scripts export server status and imported to website via php.
Banner XE Architecture

• Issues managing multiple Banner databases (continued)
  
  o Managed server application deployment tracking.
  
  • WebLogic scripts export server status and imported to website via php.

Managed Server Section

<table>
<thead>
<tr>
<th>Server Name</th>
<th>Server State</th>
<th>Server Listen Address</th>
<th>Server Listen Port</th>
<th>Server Health Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>xedev1a_1</td>
<td>RUNNING</td>
<td>bannerxeserver/paddr</td>
<td>8080</td>
<td>Component:ServerRuntime State:HEALTH_OK MBean:xedev1a_1 ReasonCode:[]</td>
</tr>
<tr>
<td>xedev1a_2</td>
<td>RUNNING</td>
<td>bannerxeserver/paddr</td>
<td>8082</td>
<td>Component:ServerRuntime State:HEALTH_OK MBean:xedev1a_2 ReasonCode:[]</td>
</tr>
<tr>
<td>xedev1a_3</td>
<td>RUNNING</td>
<td>bannerxeserver/paddr</td>
<td>8084</td>
<td>Component:ServerRuntime State:HEALTH_OK MBean:xedev1a_3 ReasonCode:[]</td>
</tr>
<tr>
<td>xedev1a_4</td>
<td>RUNNING</td>
<td>bannerxeserver/paddr</td>
<td>8086</td>
<td>Component:ServerRuntime State:HEALTH_OK MBean:xedev1a_4 ReasonCode:[]</td>
</tr>
</tbody>
</table>
Banner XE Architecture

• Issues managing multiple Banner databases (continued)

  o Managed server application deployment tracking.
    • WebLogic scripts export server status and imported to website via php.

Deployment Section

<table>
<thead>
<tr>
<th>Server Name</th>
<th>Application</th>
<th>Component Type</th>
<th>Session Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>xdev1a_1</td>
<td>Banner9DS-BANDEV</td>
<td>JDBCDataSourceRuntime</td>
<td></td>
</tr>
<tr>
<td>xdev1a_1</td>
<td>Banner9DS_BANDRMS1</td>
<td>JDBCDataSourceRuntime</td>
<td></td>
</tr>
<tr>
<td>xdev1a_1</td>
<td>Banner9SsbDS-BANDEV</td>
<td>JDBCDataSourceRuntime</td>
<td></td>
</tr>
<tr>
<td>xdev1a_1</td>
<td>StudentAdvisorSSB-BANDEV</td>
<td>WebAppComponentRuntime</td>
<td>0</td>
</tr>
<tr>
<td>xdev1a_1</td>
<td>StudentApi-BANDEV</td>
<td>WebAppComponentRuntime</td>
<td>0</td>
</tr>
<tr>
<td>xdev1a_1</td>
<td>StudentSSB-BANDEV</td>
<td>WebAppComponentRuntime</td>
<td>0</td>
</tr>
</tbody>
</table>
Banner XE Architecture

• **Issues managing multiple Banner databases (continued)**

  - Managed server application deployment tracking.
  - WebLogic scripts export server status and imported to website via php.

### Server Status Section

<table>
<thead>
<tr>
<th>Memory Status</th>
<th>Currently Allocated</th>
<th>Currently Available</th>
<th>Max Allowed</th>
<th>Percentage Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdminServer</td>
<td>500</td>
<td>195</td>
<td>500</td>
<td>39</td>
</tr>
<tr>
<td>xedev1a_1</td>
<td>4776</td>
<td>2597</td>
<td>7282</td>
<td>70</td>
</tr>
<tr>
<td>xedev1a_2</td>
<td>4749</td>
<td>1793</td>
<td>7282</td>
<td>59</td>
</tr>
<tr>
<td>xedev1a_3</td>
<td>4926</td>
<td>3616</td>
<td>7282</td>
<td>82</td>
</tr>
<tr>
<td>xedev1a_4</td>
<td>4868</td>
<td>2988</td>
<td>7282</td>
<td>74</td>
</tr>
</tbody>
</table>
# Banner XE Architecture

## Capacity planning – Future configuration

<table>
<thead>
<tr>
<th>Forms Servers</th>
<th>Self-Service Servers</th>
<th>XE Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 load balanced Red Hat 5 Servers</td>
<td>4 load balanced Red Hat 5 Servers</td>
<td>4 load balanced Red Hat 6 Servers</td>
</tr>
<tr>
<td>2 CPU’s per server</td>
<td>8 CPU’s per server</td>
<td>2 load balanced Red Hat 5 Servers</td>
</tr>
<tr>
<td>64GB RAM</td>
<td>16GB RAM</td>
<td>*initial Faculty Grade Entry implementation servers</td>
</tr>
<tr>
<td>1 Managed WebLogic Server per server.</td>
<td>1 WebLogic OHS Apache instance per server.</td>
<td>16 CPU’s per server</td>
</tr>
<tr>
<td>512MB max allocated memory per managed server</td>
<td>Each OHS instance utilizes roughly 1GB of memory.</td>
<td>6 Managed WebLogic Servers per server.</td>
</tr>
</tbody>
</table>

**Admin and Self-Service apps are split.**

***We are only partially into the Student XE product at this moment.***
Banner XE Modifications
Banner XE Modifications

- Illinois has many local modifications to Ellucian products such as Oracle Forms and Self-Service applications. This need is still required with XE technology.

- Requirements
  - Access to source code for Application Support and Developers.
  - Processes to restrict access to source code.
  - Tools to modify and test changes.
  - Processes to document and distribute changes for technical users.
Banner XE Modifications

Git Repository Setup

- **Illinois git origin server**
  - Red Hat Enterprise Linux Server release 6.7
  - ssh key exchanged with Ellucian to access source git repositories
  - Create local master branch for modification
  - Git repositories are secured with gitolite
    - ssh keys exchanged with technical team and developers to access repositories
    - Allows setting read/write access to git repositories via gitolite.conf
      
      ```
      repo banner_student_registration_app
      RW+ = user1
      R  = user2
      local-master = user3
      ```
  - Modifications to source done via IntelliJ or git bash
Banner XE Modifications

Git Repository Process

- Create bare repository from Ellucian source.
  - Create single bare repository or use script to create all available repositories.

- Create local master branch for modification.
  - Find latest version and create local master branch

- Update .gitmodules to point to local origin server

- Update project submodules.

- Clean the application. **This can be time consuming
Banner XE Modifications

Git Repository Process (continued)

• Make modifications to the code
• Compile
• Run locally to test changes.
• Include readme file with specific changes for technical teams.
• Package up and deploy to WebLogic Server.
Banner XE Modifications

Modifying XE with code changes

- Simple modification, increasing length of a field from 2 to 4

  - Example; Student Schedule App

  - Modified plugins/banner_general_common.git/src/groovy/net/hedtech/banner/general/overall/SectionCrossListSection.groovy increasing length of cross list column to 4.

    - @Column(name = "SSRXLST_XLST_GROUP", nullable = false, length = 2)

      Changed to

    - @Column(name = "SSRXLST_XLST_GROUP", nullable = false, length = 4)
Banner XE Modifications

Modifying XE with code changes

- Advanced modification, disabling Last Attend Date and Hours Attended
  - Example: Student Faculty Grade Entry
  - Functional team needed a way to ignore Last Attend Date and Hours Attended when importing Midterm Grades
  - Before modification

Before modification:

- Columns include: Term Code*, CRN*, Last Attend Date, Hours Attended
- Import Grades screen shows Last Attend Date and Hours Attended

After modification:

- Columns include: Term Code*, CRN*, Midterm Grade
- Import Grades screen shows Ignore Last Attend Date and Ignore Hours Attended

Functional team wanted to prevent the system from using Last Attend Date and Hours Attended when importing midterm grades.
Banner XE Modifications

Modifying XE with code changes

- Advanced modification, disabling Last Attend Date and Hours Attended
  - Example: Student Faculty Grade Entry
  - Functional team needed a way to ignore Last Attend Date and Hours Attended when importing Midterm Grades
  - Modified
    grails-app/controllers/net/hedtech/banner/student/utils/ImportController.groovy

Original Code Allowed Last Attend Date and Hours Attended to be mapped when importing Midterm Grades

```groovy
def columns = validCourseRosterColumns
if (session.selectedSection?.gradingType == "midterm") {
    columns = columns.findAll {
        it.key != "finalgrade" && it.key != "incompletefinalgrade" && it.key != "extensiondate"
    }
```

© 2016 ELLUCIAN. CONFIDENTIAL & PROPRIETARY | Session 36090
Banner XE Modifications
Modifying XE with code changes

- **Advanced modification, disabling Last Attend Date and Hours Attended**
  - Example; Student Faculty Grade Entry
  - Functional team needed a way to ignore Last Attend Date and Hours Attended when importing Midterm Grades
  - After modification
Banner XE Modifications

Modifying XE with code changes

- Advanced modification, disabling Last Attend Date and Hours Attended
  - Example; Student Faculty Grade Entry
  - Functional team needed a way to ignore Last Attend Date and Hours Attended when importing Midterm Grades
  - Modified
    ```groovy
    grails-app/controllers/net/hedtech/banner/student/utils/ImportController.groovy
    ```

Modified Code Ignored Last Attend Date and Hours Attended mappings when importing Midterm Grades by adding lastattendeddate and hoursattended columns to logic. Change highlighted in red.

```groovy
def columns = validCourseRosterColumns
if (session.selectedSection?.gradingType == "midterm")
columns = columns.findAll {
  it.key != "finalgrade" && it.key != "incompletefinalgrade" && it.key != "extensiondate" && it.key != "lastattendeddate" && it.key != "hoursattended"
}
```
Banner XE Modifications

Modifying XE without code changes

- **Modifying XE through CSS and js**
  - Include bannerSelfService-custom.css and bannerSelfService-custom.js in XE war.

  ```
  .institutionalBranding {
    background: url("./images/logo.gif") no-repeat;
  }
  ```

  - Allows for Institutional Branding, removing the Ellucian University logo in upper left of browser.

  - Allows hiding of columns/buttons without modifying source code. Example, Student Facult Grade Entry

    ```
    #browseButton {
      visibility: hidden;
    }
    ```

  - Using browser developer tools makes finding elements easier.
Banner XE Modifications

Modifying XE without code changes

- Modifying XE through CSS and js

Before custom css
Banner XE Modifications

Modifying XE without code changes

- Modifying XE through CSS and js

After custom css
Banner XE Modifications

Git/Grails Challenges

- Learning and integrating new modification process into existing procedures
- Maintaining changes to application plugins that are used by multiple XE applications
- Windows specific issues
  - Some groovy files needed to be changed to accommodate Windows paths. “grails not found”
  - Setting fileformat=unix when editing some files
- Local workstation resources when cleaning, compiling, and running apps locally. Need a good amount of RAM if workstation isn’t dedicated to modifications.
Tips/Troubleshooting XE
Skills needed to support XE

- Existing expert knowledge of Banner 8
- Java => Groovy and Grails
- Web 2.0 (HTML5, CSS, JavaScript, AJAX, jQuery, JSON)
- Java Middle Tier (tomcat or WebLogic)
- Oracle database
- Git (source control)
- IDE (IntelliJ or Eclipse)
- Persistence and time…
Resources

- eCommunities/BORACLE
- Get involved in early adopter/beta efforts (good for contacts)
- Literature (Grails in Action/Groovy in Action)
- Ellucian training on XE development
- Lot of free resources in terms of Git, HTML5, Javascript, JQuery and Java
- Groovy and Grails is a little spotty on google
Browsers

• Works well with Chrome and Firefox (haven’t tested much with Safari)
• Don’t recommend IE. IE8 JavaScript was REALLY slow. Faster now but have seen issues with things not displaying properly
• Keeping in mind that Chrome and Edge (cannot run Banner 8 Forms). Firefox targeting end of the year.
Browsers for troubleshooting

- Chrome Developer Tools are your friend!! 🎁 (View-> Developer -> Developer Tools)
- Assists with performance, response headers, CSS manipulation, cookies, errors, etc..
- For Firefox there is (Tools -> Web Developer)
- Other tools include Firebug, Fiddler, SAML tracer (setting up SAML), etc…
- Something for every browser
Chrome Dev Tools

1.) Click on Search Terms Drop Down

2.) Review Background AJAX calls

3.) Evaluate Performance
Chrome Dev Tools cont...

1.) Highlight “getTerms” request

2.) Click on “Preview” tab then view payload from “getTerms” request
Finding source code from url

URL: https://host/StudentRegistrationSSB/ssb/classRegistration/getTerms

def getTerms():
    if session.getAttribute("selectedRole")?.administrator
        terms = selfServiceLookupService.fetchSelectableTermsForRegistrar(
            params.offset.toInteger() + params.max.toInteger() *
            params.offset.toInteger())
    else:
        terms = selfServiceLookupService.fetchSelectableTerms(
            params.offset.toInteger() + params.max.toInteger() *
            params.offset.toInteger())
    render terms.collect {
        [code: it.code, description: it.description.encodeAsHTML()]
    } as JSON
Chrome Dev Tools CSS fun!!

1.) Goal – modify the appearance
2.) Click “Elements” tab and highlight “select2-results” class for example
3.) Elements on page automatically highlight and css shows up on right
Chrome Dev Tools CSS fun cont..

1.) Click Add

2.) Add some css

3.) Changes magically appear!!
Add css changes to XE app

- Create or modify $BANNER_HOME/xe_app/instance/config/css/bannerSelfService-custom.css file
- `ul#select2-results {
  color: red;
}

- Save the changes and deploy
- We’ve done this to effectively hide fields, columns, etc.
Application Log File

- Comes with every application and can be customized in terms filename/directory/logging level
- First place to start with issues
- Inside file StudentRegistrationSsb_configuration.groovy

```groovy
log4j = {
    def String loggingFileDir = "/u01/app/oracle/Apache_logs"
    def String logAppName = "StudentRegistrationSSB-BANDR1"
    def String loggingFileName = "${loggingFileDir}/${logAppName}.log".toString()
    appenders {
        rollingFile name:'appLog', file:loggingFileName, maxFileSize:"${10*1024*1024}", maxBackupIndex:10, layout:pattern(conversionPattern: '%d{[EEE, dd-MMM-yyyy @ HH:mm:ss.SSS]} [%t] %-5p %c %x - %m%n')
    }
}
```

- custom logging directory
- custom log file name
Application Log Error

Specific error: MecCodeNotFoundException occurred when processing request: [GET] /StudentRegistrationSSB-BANDR1/ssb/registration

No Institution code specified in the URL. Stacktrace follows:

```
at net.hedtech.banner.db.BannerDS.setMepSsb(BannerDS.groovy:642)
at net.hedtech.banner.db.BannerDS.getConnection(BannerDS.groovy:81)
at net.hedtech.banner.student.registration.RegistrationController.getGtvsdaxForSession(RegistrationController.groovy:66)
at net.hedtech.banner.student.registration.RegistrationController.registration(RegistrationController.groovy:18)
at weblogic.servlet.internal.StubSecurityHelper$ServletServiceAction.run(StubSecurityHelper.java:227)
```
XE Application debugging

• XE comes with log4j built into it
• Can modify logging level across board (careful with VERY large log files with trace or debug)
• Can enable logging for individual packages or classes
• Can customize log file name and directory
• Changes made to [appname]_configuration.groovy then rebuild and deploy
• JMX extensions for dynamic logging (haven’t tried yet)
• Customized logging by modifying code/rebuild/redeploy
Global Log Change

- Global log change using root logger in [appname]_configuration.groovy
- Log levels (off, fatal, error, warn, info, debug, trace, all)

```groovy
case 'production':
    root {
        error 'appLog'
        additivity = true
    }
```

Output to defined appender or default “stdout”
Targeted log change

e.g. having authentication issues??

// ******* Security framework classes **********

trace 'net.hedtech.banner.security'
trace 'net.hedtech.banner.db'
trace 'net.hedtech.banner.security.BannerAccessDecisionVoter'
trace 'net.hedtech.banner.security.BannerAuthenticationProvider'
off 'net.hedtech.banner.security.CasAuthenticationProvider'
trace 'net.hedtech.banner.security.SelfServiceBannerAuthenticationProvider'
trace 'org.grails.plugins.springsecurity'
trace 'org.springframework.security'
trace 'org.apache.http.headers'
trace 'org.apache.http.wire'
Custom log change

Using getTerms again as example, add following sample code to ClassRegistrationController.groovy then run locally

```groovy
add debug code

log.error "** log test for eLive 2016 **"
log.error "${terms}"
log.error "** end log test for eLive 2016"
```
Custom logging output

- View output in log or stdout
- Output of “terms” is nicely displayed automatically

Configuring Spring Security Core ...

... finished configuring Spring Security Core

| Server running. Browse to http://localhost:8080/StudentRegistrationSsb
Oracle AWR reporting

- AWR (Automated Workload Repository)
- Note – does require diagnostics pack license
- Database snapshots taken every hour by default (or manually)
- The awrrpt.sql SQL script generates an HTML or text report that displays statistics for a range of snapshot IDs.
- I like to isolate in a dev/test environment and run something
- Essential for use in load testing
- Great for “generically” tracing a connection pool
AWR in action/Reg XE Load testing

Disaster!!!

**Top 10 Foreground Events by Total Wait Time**

<table>
<thead>
<tr>
<th>Event</th>
<th>Waits</th>
<th>Total Wait Time (sec)</th>
<th>Wait Avg(ms)</th>
<th>% DB time</th>
<th>Wait Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>latch: row cache objects</td>
<td>13,754,847</td>
<td>120.4K</td>
<td>9</td>
<td>79.9</td>
<td>Concurrency</td>
</tr>
<tr>
<td>library cache: mutex X</td>
<td>29,846,296</td>
<td>21.1K</td>
<td>1</td>
<td>14.0</td>
<td>Concurrency</td>
</tr>
<tr>
<td>DB CPU</td>
<td>9554</td>
<td></td>
<td></td>
<td>6.3</td>
<td></td>
</tr>
<tr>
<td>cursor: pin S wait on X</td>
<td>620</td>
<td>52.3</td>
<td>84</td>
<td>0.0</td>
<td>Concurrency</td>
</tr>
<tr>
<td>cursor: mutex X</td>
<td>158,431</td>
<td>50.1</td>
<td>0</td>
<td>0.0</td>
<td>Concurrency</td>
</tr>
</tbody>
</table>

**BEFORE**

**AFTER**

JDBC Caching
AWR features

<table>
<thead>
<tr>
<th>Snap Id</th>
<th>Snap Time</th>
<th>Sessions</th>
<th>Cursors/Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin Snap:</td>
<td>729</td>
<td>09-Nov-15 08:31:22</td>
<td>220</td>
</tr>
<tr>
<td>End Snap:</td>
<td>730</td>
<td>09-Nov-15 08:06:16</td>
<td>222</td>
</tr>
<tr>
<td>Elapsed:</td>
<td>34.89 (mins)</td>
<td>34.88 (mins)</td>
<td></td>
</tr>
<tr>
<td>DB Time:</td>
<td>34.88 (mins)</td>
<td>34.88 (mins)</td>
<td></td>
</tr>
</tbody>
</table>

SQL ordered by elapsed time

- SQL ordered by Elapsed Time
- SQL ordered by CPU Time
- SQL ordered by User I/O Wait Time
- SQL ordered by Gets
- SQL ordered by Reads
- SQL ordered by Physical Reads (UnOptimized)
- SQL ordered by Executions
- SQL ordered by Parse Calls
- SQL ordered by Shared Memory
- SQL ordered by Version Count
- Complete List of SQL Text

AWR header

SQL report options

2 queries 67% of activity!!

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 113.5% of Total DB Time (s): 2.031
- Captured PL/SQL account for 22.5% of Total DB Time (s): 2.091

<table>
<thead>
<tr>
<th>Elapsed Time (s)</th>
<th>Executions</th>
<th>Elapsed Time per Exec (s)</th>
<th>%Total</th>
<th>%CPU</th>
<th>%IO</th>
<th>SQL ID</th>
<th>SQL Module</th>
<th>SQL Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>891.06</td>
<td>1</td>
<td>891.06</td>
<td>42.86</td>
<td>11.50</td>
<td>70.74</td>
<td>RPE16G3</td>
<td>select ((SUBSTR(SPIDEN_LAST...</td>
<td></td>
</tr>
<tr>
<td>497.36</td>
<td>703</td>
<td>0.69</td>
<td>23.91</td>
<td>44.05</td>
<td>5.30</td>
<td>RPE16G3</td>
<td>select X.RPRASS6.FUND_CODE.K...</td>
<td></td>
</tr>
<tr>
<td>336.35</td>
<td>763</td>
<td>0.44</td>
<td>16.16</td>
<td>42.66</td>
<td>59.11</td>
<td>RPE16G3</td>
<td>declare LV_PARM_NEXT_UP number...</td>
<td></td>
</tr>
<tr>
<td>277.86</td>
<td>255,161</td>
<td>0.00</td>
<td>13.36</td>
<td>8.46</td>
<td>0.86</td>
<td>RPE16G3</td>
<td>SELECT RGRAFM.C_INST_.1_TOT,f...</td>
<td></td>
</tr>
<tr>
<td>120.88</td>
<td>2,905</td>
<td>0.04</td>
<td>5.61</td>
<td>54.76</td>
<td>40.45</td>
<td>RPE16G3</td>
<td>SELECT SFRSTOR_CRN, SFRSTOR_TE...</td>
<td></td>
</tr>
<tr>
<td>118.16</td>
<td>2,805</td>
<td>0.04</td>
<td>5.66</td>
<td>10.25</td>
<td>91.37</td>
<td>RPE16G3</td>
<td>SELECT SHRTCKN_CRN, SHRTCKN_TE...</td>
<td></td>
</tr>
</tbody>
</table>
Summary

- It’s a luxury to be able to use Banner 8 while you get ramped up on XE (now’s the time!!)
- Ease your way into it/get comfortable with the technologies and architecture – there is a learning curve
- Make some mods if only for the practice of seeing it all put together
Thank you!

Brian Schoudel – brians@uillinois.edu
John Tanner – jrtanner@uillinois.edu

Please complete the online session evaluation form.
Session ID 36090
We need your help

Feedback Survey

Access session surveys by using the survey widget on the mobile app or by logging into your session schedule builder at

http://tinyurl.com/elive2016surveys

Session ID 36090