

# CAS & Terracotta

A review of NAU's implementation of CAS  
& Terracotta at an enterprise level

# Overview

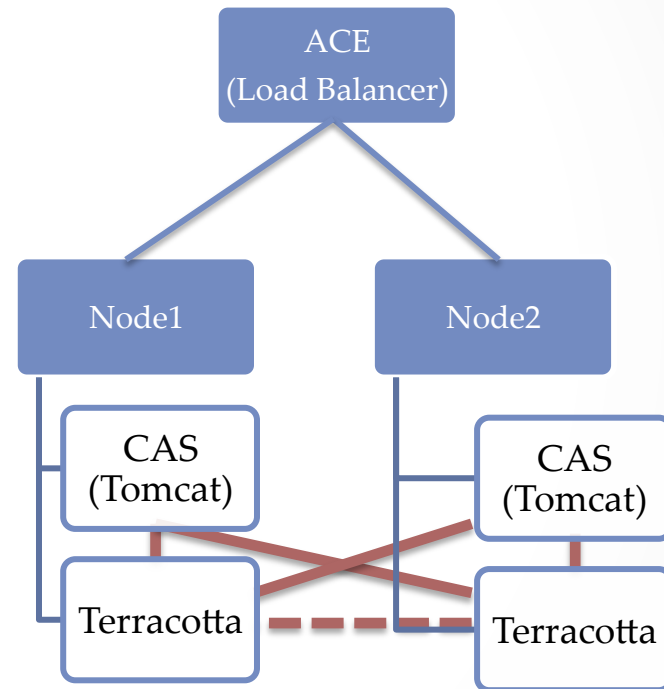
- A Background on CAS @ NAU
- Terracotta & CAS Setup
- Terracotta Configuration
- Java Memory Tuning
- Improvements & The Future

# CAS @ NAU Background

- Major Services
  - PeopleSoft ~100k accts
  - Shibboleth
    - GAE (Google Apps)  
~90k accts 45k active
    - Confluence
  - uPortal 100k accts 29k active
  - WebCT Vista ~60k accts
  - Business Objects
  - Various Websites
- Clients
  - Homebrewed with additional authorization layer (NAUauthZ)
  - Java
  - Perl (Apache)
  - .Net
- Versions & Additions
  - CAS 3.4.2
    - Modifications for old style “PT” support
  - Password Expiration Module
- Statistics  
(very rough numbers)
  - Logins per hour  
1000 – 10000
  - Logins per day ~100k
  - Tickets granted per hour  
1000 – 22000

# Terracotta & CAS Setup

- Why Terracotta?
  - Jboss not stable enough
  - Simple implementation
  - Improved redundancy
  - Developer Console
- Setup
  - Hardware
    - Cisco ACE (Load Balancer)
    - 2 quad-processor dual core AMD2200 nodes w/ 8Gb memory
  - Software
    - Terracotta 3.2.1\_1
    - Tomcat 5.5.27
    - Apache (SSL handling)
    - CAS 3.4.2
      - Default Ticket Registry



# Terracotta Configuration

- “Simple”

All configuration is done via “tc-config.xml” & a few commands

- DSO vs Express:

CAS was initially suggested to be run on Terracotta DSO (Distributed Shared Objects) which made more sense

<http://www.theserverside.com/news/1363891/Using-Terracotta-DSO>

<http://www.infoq.com/articles/open-terracotta-intro>

- CAS Integration:

Terracotta requires modules to be installed:

<http://www.terracotta.org/confluence/display/docs/Integrating+Terracotta+DSO>

- tim-vector
- tim-tomcat-5.5
- tim-spring-webflow-2.0
- tim-spring-security-2.0

- Initial configuration notes:

<https://issues.jasig.org/browse/CAS-750>

# TC Configuration cont.

- Application (CAS) awareness
  - Defined in <application> section of tc-config.xml
    - <roots> section contains configuration of shared memory roots
    - <locks> section contains configuration for memory locks in individual methods.
    - <instrumented-classes> any object referenced by a shared memory object needs to be listed here.
- Multiple node awareness
  - Defined in <servers> section of tc-config.xml
  - One server entry for each node
- Failover awareness
  - Defined in <tc-properties> section
  - Reference:  
<http://www.terracotta.org/confluence/display/wiki/TUNING+TERRACOTTA#TUNINGTERRACOTTA-FAILOVER%3A>

# Java Memory Tuning

- Memory!
  - The majority of our tuning has been taken from our prior experiences, mostly from our Shibboleth Terracotta & uPortal setups.
  - Tomcat (catalina.sh)
    - `JAVA_OPTS="$JAVA_OPTS -Xmx1024M -XX:MaxPermSize=1024M -Dtc.install-root=/local/path/terracotta-3.2.1_1 -Dtc.config=/local/path/tc-config.xml -Xbootclasspath/p:/local/path/terracotta-3.2.1_1/lib/dso-boot/dso-boot-hotspot_linux_160_13.jar"`
  - Terracotta (start-tc-server.sh)
    - `SERVER_OPTS="$SERVER_OPTS -Xms1024m -Xmx1024m -XX:NewRatio=3 -XX:MaxTenuringThreshold=15 -XX:+HeapDumpOnOutOfMemoryError -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:+UseParallelOldGC -XX:MaxGCPauseMillis=5000 -XX:-TraceClassUnloading"`
  - Reference:  
<http://java.sun.com/performance/reference/whitepapers/tuning.html>

# Improvements & The Future

- Issues
  - Moved to dedicated Tomcat for CAS
- Memory management & garbage collection
  - Always room for improvement in garbage collection
  - Is BigMemory worth it?
- Terracotta Updates
  - On the horizon...
- Dedicated NICs for Terracotta