A Technical Analysis of the September 2010 Partial SSO Interruptions

- Problem
- Immediate Workaround
- Cause
- Resolution
- Additional Action Items

**Problem**

UCLA's Shibboleth SSO service experienced four episodes of partial service interruptions between September 21 and September 30, 2010. These events were:

- 9/21 1 - 7 AM
- 9/22 12 - 12:10 PM
- 9/24 4:30 - 7:30 AM
- 9/30 5:01 - 6:45 AM

In each case, one of our load balanced servers ceased to respond to user traffic, causing a portion of the users to not able to sign into campus SSO-enabled web applications.

**Immediate Workaround**

In each of the above instances, as soon as we became aware of the problem we took the problematic server out of the load balancer, restarted the web server and reintroduced the server to the load balancer pool.

**Cause**

The Shibboleth Identity Provider (IDP) 2.2 software uses Terracotta to provide user session load balancing. During operation, client nodes (in this case, Shibboleth IDP) write session data to Terracotta. Terracotta monitors the health of its clients by periodically pinging its clients. If Terracotta fails to receive responses from a client for a prolonged period of time, it disconnects the client.

Unfortunately, when Shibboleth is disconnected from Terracotta due to such timeout, it becomes unstable and virtually unresponsive. The web server continues to accept requests from the load balancer, but can't process them. This behavior causes requests to queue and eventually results in a outage.

We have determined that these timeouts were caused by longer than normal Java virtual machine (JVM) garbage collection (GC). In any Java application, the JVM periodically performs garbage collection to reclaim unused memory. JVM garbage collection is a complicated and potentially resource intensive operation. Many factors influence how long the GC process runs and whether it is disruptive to normal processes. These factors include the number of processor available, the amount of memory allocated, the garbage collection algorithm used, and other tuning parameters. In our cases, the garbage collection processes ran longer than usual, lasting up to a minute. The long running GC process blocked Shibboleth and the web server from responding to Terracotta's pings. As a result, Terracotta disconnected the unresponsive client, causing Shibboleth to go into a sustained unresponsive state.

**Resolution**

We tuned JVM Garbage collection algorithm to be more efficient and non-disruptive. We believe this greatly reduces chances of a similar outage.

In addition, we also brought online an additional server to reduce the amount of web traffic going to each server.

**Additional Action Items**

**Tuning VM and web servers**

We are continuing to further tune the JVM, Terracotta, as well as the Virtual Machine configurations.
Improving Notification

We are looking to add additional, finer grained monitoring processes to notify us of not only existing, but impending failures.

We are working to improve our notification procedure so that the on call support personnel are contacted as soon as problem occurs.