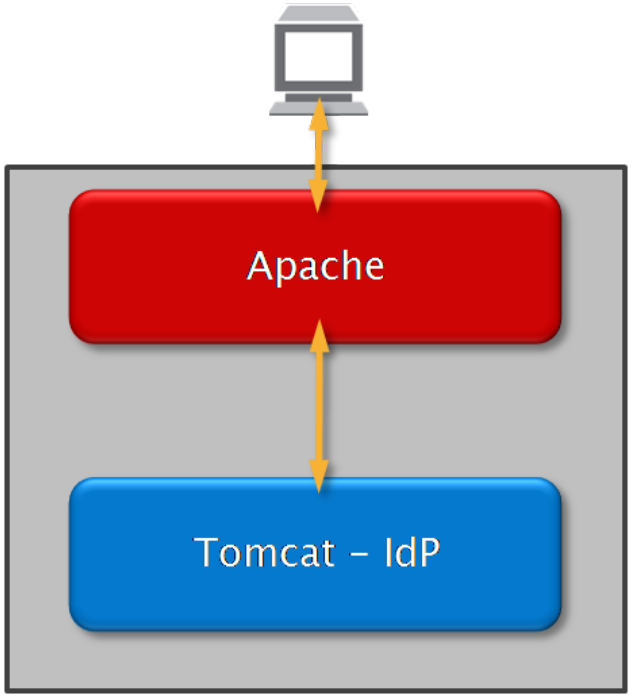
IDP Shibboleth in cluster

Fabio Spelta -Università degli Studi di Milano-Bicocca

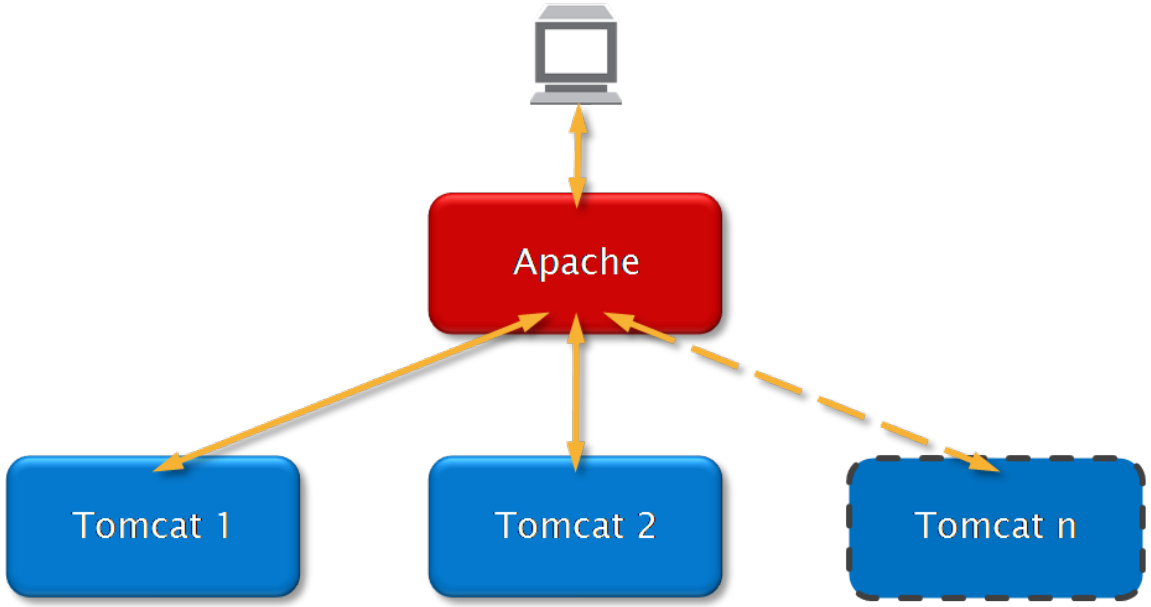
Installazione classica di Shibboleth

Apache e Tomcat, spesso su un solo sistema



Primo approccio: replicare gli IDP

Si "fotocopiano" i sistemi con Tomcat
Apache continua a fare da reverse proxy bilanciando il carico



Bisogna tenere traccia delle sessioni esistenti!

Replicare gli IDP - configurazione di Tomcat

```
<Engine name="Catalina" defaultHost="
localhost" jvmRoute="shib2">
```

Esempio di cookie:

```
Name: JSESSIONID
Content: 30C39B5E0752B26D8E704985C06D3B7A.
shib2
Host: idp.unimib.it
Path: /idp/
Send for: Encrypted connections only
Expires: At end of session
```

Replicare gli IDP - configurazione di Apache

- Load balancer

```
<Proxy balancer://clusteridp>
     BalancerMember ajp://10.0.0.1:8009 route=shib1
     BalancerMember ajp://10.0.0.2:8009 route=shib2
     BalancerMember ajp://10.0.0.3:8009 route=shib3
     ProxySet stickysession=JSESSIONID
     Allow from All
</Proxy>
```

- Reverse Proxy

```
RewriteEngine On
RewriteRule ^/idp/(.*) balancer://clusteridp/idp/$1   [P,L]
RewriteRule    .*   -  [F]
```

- Reverse Proxy magic

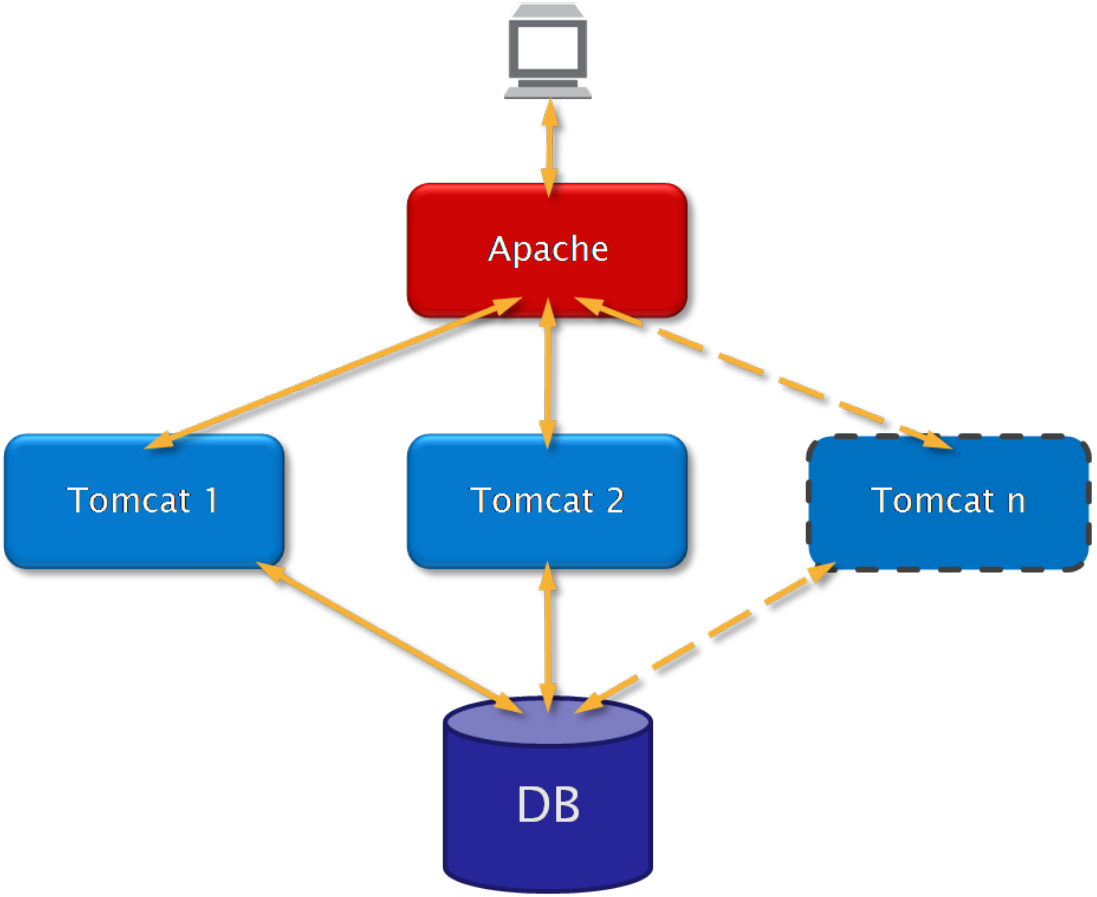
```
RewriteCond %{REMOTE_ADDR} ^149\.132\.5\.240$
RewriteRule ^/idp/(.*) ajp://10.0.0.1:8009/idp/$1  [NC,P,L]
RewriteRule ^/idp/(.*) balancer://cluster_idp_due_e_tre/idp/$1  [NC,P,L]
```

I nodi sono *quasi* del tutto indipendenti

- A meno che non si usi **storedId** per persistentId (il che tutto sommato è una buona idea)

	LOCALENTITY	PEERENTITY	PRINCIPALNAME	LOCALID	PERSISTENTID
10984	https://idp.unimib.it/idp/shibboleth	http://nera.cineca.it/simplesa...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10985	https://idp.unimib.it/idp/shibboleth	http://shibboleth.ebscohost.c...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10986	https://idp.unimib.it/idp/shibboleth	https://aai.caspur.it/shibboleth	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10987	https://idp.unimib.it/idp/shibboleth	https://sdauth.sciencedirect....	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10988	https://idp.unimib.it/idp/shibboleth	https://wiki.si.unimib.it/simpl...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10989	https://idp.unimib.it/idp/shibboleth	https://ricevi.ct.infn.it/shibbol...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10990	https://idp.unimib.it/idp/shibboleth	https://ieeexplore.ieee.org/s...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10991	https://idp.unimib.it/idp/shibboleth	google.com/a/campus.unimib.it	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10992	https://idp.unimib.it/idp/shibboleth	https://wims2.matapp.unimib...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10993	https://idp.unimib.it/idp/shibboleth	https://www.annualreviews.o...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10994	https://idp.unimib.it/idp/shibboleth	https://foodl.org/simplesaml/...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10995	https://idp.unimib.it/idp/shibboleth	https://filesender.garr.it/sim...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]
10996	https://idp.unimib.it/idp/shibboleth	https://sp2-test.garr.it/shibb...	fabio.spelta@unimib.it	fabio.spelta@unimib.it	[illegible]

Apache + *n* Tomcat + DB



Il DB può essere facoltativamente "clusterizzato" a sua volta!

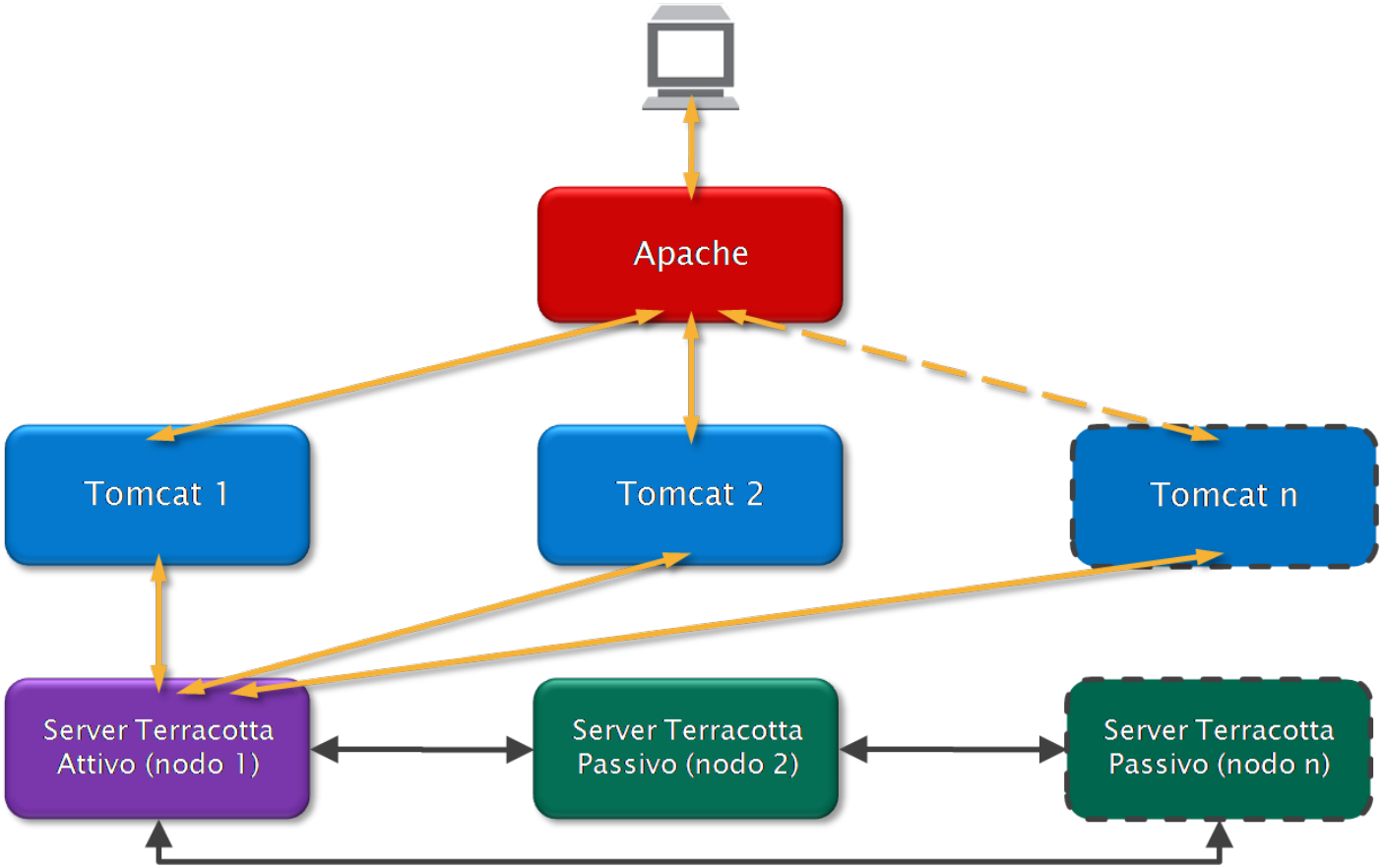
Accesso concorrente al DB

- Oracle e MySQL gestiscono automaticamente le scritture concorrenti (con dei lock sulla tabella)
- Occorre comunque creare le chiavi per garantire l'univocità dei record

```
KEY persistentId (persistentId),
KEY persistentId_2 (persistentId, deactivationDate),
KEY localEntity (localEntity(16), peerEntity(16), localId),
KEY localEntity_2 (localEntity(16), peerEntity(16), localId,
deactivationDate)

CREATE UNIQUE INDEX persistentId ON shibpid(persistentId);
CREATE UNIQUE INDEX persistentId_2 ON shibpid(persistentId,
deactivationDate);
CREATE UNIQUE INDEX localentity ON shibpid(localEntity,
peerEntity, localId);
CREATE UNIQUE INDEX localentity_2 ON shibpid(localEntity,
peerEntity, localId, deactivationDate);
```

Altra carne (anzi, terra) al fuoco



Terracotta sì

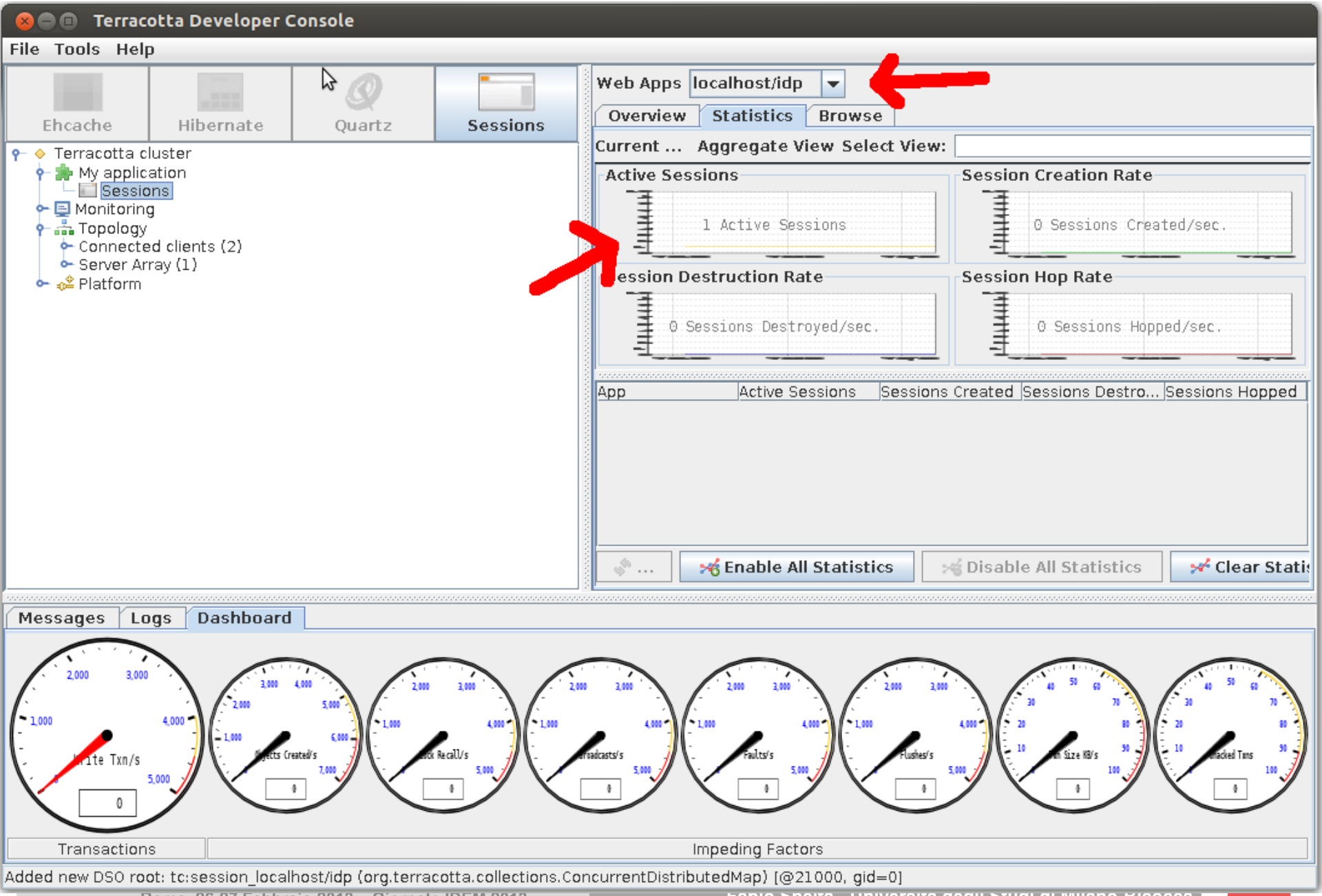
- Se un nodo IdP fallisce, gli utenti che hanno una sessione attiva presso quel nodo non devono riautenticarsi prima del termine della stessa
  - Se si usa CAS o un altro sistema di login SSO esterno, il problema non si pone comunque

- Si può continuare a supportare SAML1
  - volendo si può fare anche senza Terracotta, ma *non* in cluster

- **Se un nodo IdP fallisce *mentre* un utente si autentica la sessione viene gestita da un altro nodo, altrimenti si guasta**

Terracotta no

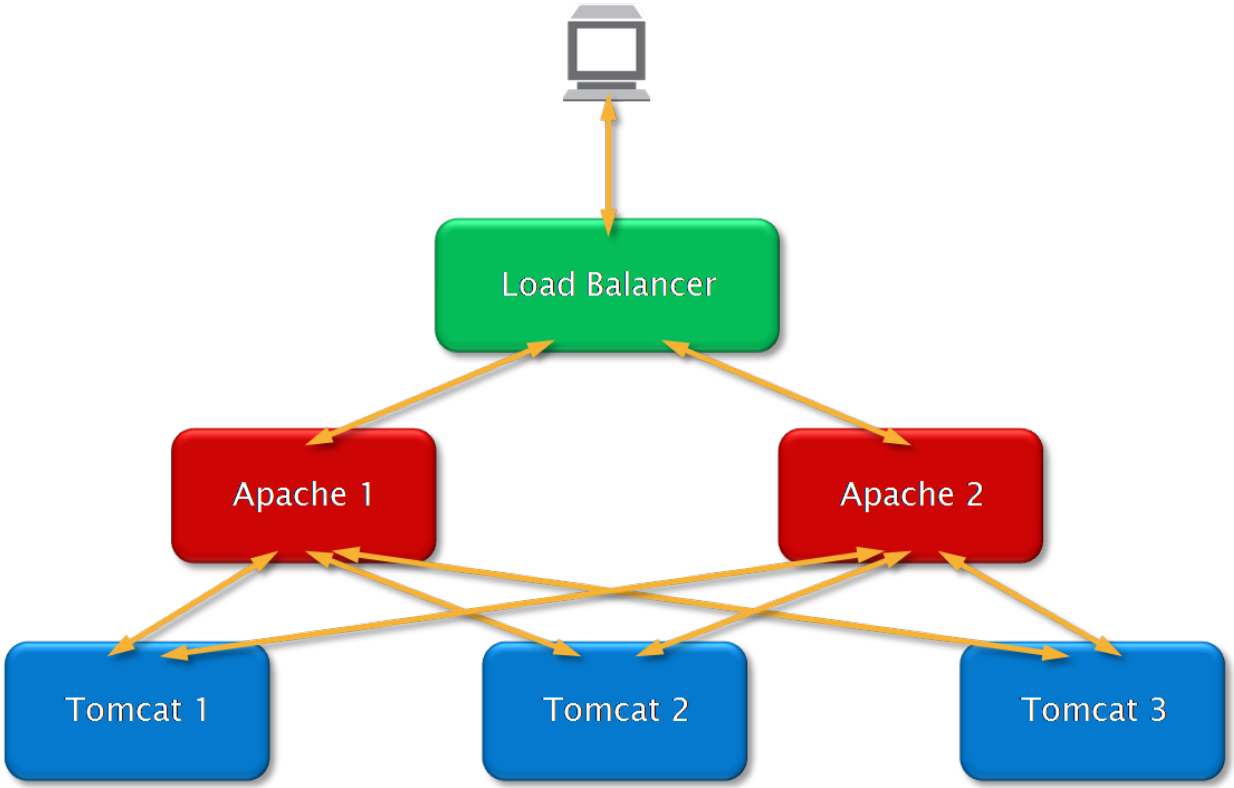
- Terracotta **non supporta più Tomcat in modalità DSO** dalla versione 3.7.2 (siamo alla 3.7.4)
- DSO vs Express:
- Con DSO (3.7.0 e precedenti):
  - "Under certain circumstances, standbys may fail to automatically clear their data directory and fail to restart, generating errors. In this case, the data directory must be manually cleared."
  - "Even if the standby's data is cleared, a copy of it is saved. By default, the number of copies is unlimited. Over time, and with frequent restarts, these copies may consume a substantial amount of disk space if the amount of shared data is large. You can manually delete these files"
  - Suppora solo Java 6 e il supporto di Sun/Oracle per Java 6 si esaurisce a febbraio 2013! Terracotta 3.7.3 è la prima versione a supportare Java 7 (ma non Tomcat via DSO) http://terracotta.org/confluence/display/release/Terracotta+3.7%2C+Ehcache+2.6%2C+Quartz+2.1+Platform+Support
- Con Express:
  - apparentemente non funziona...


Terracotta Developer Console
File Tools Help
Ehcache
Hibernate
Quartz
Sessions
Terracotta cluster
My application
Sessions
Monitoring
Topology
Connected clients (2)
Server Array (1)
Platform
Web Apps localhost/idp
Overview Statistics Browse
Current ... Aggregate View Select View:
Active Sessions
1 Active Sessions
Session Creation Rate
0 Sessions Created/sec.
ession Destruction Rate
0 Sessions Destroyed/sec.
Session Hop Rate
0 Sessions Hopped/sec.
App Active Sessions Sessions Created Sessions Destro... Sessions Hopped
Enable All Statistics
Disable All Statistics
Clear Statis
Messages Logs Dashboard
Transactions
Impeding Factors
Added new DSO root: tc:session_localhost/idp (org.terracotta.collections.ConcurrentDistributedMap) [@21000, gid=0]

Morale?

- Bisognerebbe usare una versione di Terracotta già obsoleta (ammesso di riuscire a scaricarla)
- ...e oltretutto non priva di problematiche (manutenzione manuale)

Server party: moltiplicare anche i frontend



- Anche i frontend sono "fotocopie" e sono indipendenti l'uno dall'altro
- Bilanciati a livello superiore (tipicamente in hardware)

Ringraziamenti

- Valzorio (CINECA) per il tempo dedicatoci con SSP
- Manfredi e Zanmarchi (unipd) per averci ~~fatto copiare tutte le configurazioni~~ ispirato questa soluzione
- Tordini (unimib) per tutto il resto

Appendice: Possibili alternative a Shibboleth?

- Ricerca di possibili alternative sulla base di alcune caratteristiche ritenute indispensabili per l'attuale implemetazione dell'IdP unimib:
  - Load balancing applicativo
  - Client CAS per l'autenticazione degli utenti
  - Prelievo attributi da server LDAP
  - Filtri sugli attributi

- Tra i candidati possibili, ***Authentic** 2* e ***Josso*** mancavano del componente client per l'autenticazione CAS.

- **SimpleSAMLPHP** risponde a tutti i requisiti, ma...

SimpleSAMLPHP

- Vantaggi:
  - rapidità di installazione, sia per la parte SSP che per la parte memcached
  - gestione esclusiva delle dipendenze tramite repository CentOS (anche memcached)
  - elevata possibilità di personalizzazione

- Svantaggi:
  - confezionato per federazione dove gli SP espongono gli attributi obbligatori nei metadati (come Feide); altrimenti, richiesta personalizzazione per il rilascio degli attributi (e.g. modulo AttributePolicy by Valzorio ©)
  - nonostante il modulo AttributePolicy, rimangono problemi nella valorizzazzione di ePSA: è necessario rilasciare *l'attributo sorgente* agli SP per poter essere processato dall'*authproc filter*; ciò implica che debba essere "svuotato" (*unset*) per evitare che in ultima istanza arrivi allo stesso SP (chiaro, vero?)

Domande?