



Кластеризация *Apache Tomcat*



Михаил Сильванович
(Michael_Silvanovich at epam.com)



Agenda

- Кластеризация
- Репликация сессий. Зачем? Когда применять?
- Репликация сессий. Возможные конфигурации
- Q&A



Кластеризация

Кластеризация. Балансировка.

Failover. Репликация сессий.



Кластеризация

- Масштабируемость
- Управление состоянием
- Балансировка нагрузки
- Высокая доступность:
 - на уровне запроса
 - *на уровне сессии*



Управление состоянием

- Stateless-приложения (без сессий)
- Stateful-приложения
 - варианты:
 - HTTP-сессия
 - БД
 - параметры запроса
- Влияет в общем случае на алгоритм балансировки



Управление состоянием

- Sticky-сессии
 - идентификатор сессии
 - cookie
 - некоторый уникальный атрибут клиента (например, IP-адрес)*

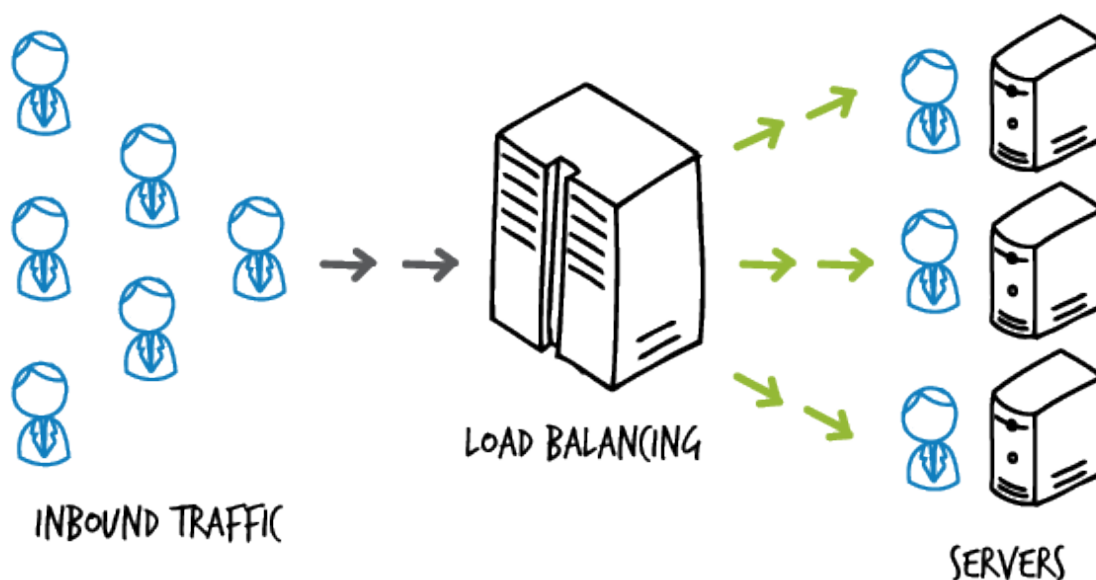


Кластеризация

- Масштабируемость
- Управление состоянием
- **Балансировка нагрузки**
- Высокая доступность:
 - **на уровне запроса**
 - *на уровне сессии*



Балансировка нагрузки





Балансировка нагрузки

- Hardware balancer
- Software balancer:
 - httpd + mod_proxy
 - Nginx
 - JK-балансировщик:
 - apache2 + **mod_jk**
 - apache2 + **mod_proxy_ajp**



Кластеризация

- Масштабируемость
- Управление состоянием
- Балансировка нагрузки
- Высокая доступность:
 - на уровне запроса
 - *на уровне сессии*



Репликация сессий. Зачем? Когда применять?

- В идеале – НИКОГДА
 - усложняется конфигурация
 - вносит дополнительный overhead
 - усложняется отладка
- Что действительно необходимо?
 - балансировка + Sticky-сессии
 - мертвая нода = мертвая сессия



Репликация сессий. Зачем? Когда применять?

- Возможно, Вашу проблему решает **Parallel Deployment**?



Репликация сессий. Зачем? Когда применять?

- Возможно, Вашу проблему решает **Parallel Deployment**?
- По-прежнему надо?



Репликация сессий

- Persistent Manager
- SimpleTcpCluster
 - DeltaManager
 - BackupManager
- Стороннее решение
 - Memcached
 - Hazelcast



Репликация сессий

- **Persistent Manager**
 - Persistent Store:
 - File Store
 - JDBC Store
 - Конфигурация:
 - Min/Max Idle Backup Time (sec)
 - Min/Max Idle Swap Time (sec)



Per

- Per

- P

- K

```
michael@michael-Lenovo-G580: ~/Sandbox/projects/clustering/persistent/tomcat-docker/
<?xml version='1.0' encoding='utf-8'?>
<Context>

  <!-- Default set of monitored resources -->
  <WatchedResource>WEB-INF/web.xml</WatchedResource>

  <Resource name="jdbc/sessions"
    auth="Container"
    type="javax.sql.DataSource"
    username="postgres"
    password="postgres"
    driverClassName="org.postgresql.Driver"
    url="jdbc:postgresql://sql.hardcoders.ru:5432/tc"
    maxActive="20"
    maxIdle="10"
    validationQuery="select 1" />

  <Manager className="org.apache.catalina.session.PersistentManager"
    distributable="true"
    processExpiresFrequency="3"
    maxIdleBackup="1"
    maxIdleSwap="5"
    minIdleSwap="-1"
    saveOnRestart="true" >

    <Store className="org.apache.catalina.session.JDBCStore"
      dataSourceName="jdbc/sessions"
      sessionTable="tomcat_sessions"
      sessionIdCol="session_id"
      sessionAppCol="app_name"
      sessionDataCol="session_data"
      sessionLastAccessedCol="last_access"
      sessionMaxInactiveCol="max_inactive"
      sessionValidCol="valid_session" />

  </Manager>

</Context>
```




Репликация сессий

- **Persistent Manager**
 - Persistent Store:
 - File Store
 - JDBC Store
 - Конфигурация:
 - Min/Max Idle Backup Time (sec)
 - Min/Max Idle Swap Time (sec)
 - Не гарантирует failover



Репликация сессий

- ~~Persistent Manager~~
- SimpleTcpCluster
 - DeltaManager
 - BackupManager
- Стороннее решение
 - Memcached
 - Hazelcast





Репликация сессий

- SimpleTcpCluster
 - DeltaManager
 - BackupManager
- Стороннее решение
 - Memcached
 - Hazelcast



Репликация сессий

- **SimpleTcpCluster**
 - DeltaManager
 - BackupManager



Репликация сессий

- SimpleTcpCluster
 - Компоненты
 - **Cluster**
 - контейнер для всех компонентов кластеризации
 - может быть объявлен как в Engine, так и в Host



Репликация сессий

- SimpleTcpCluster
 - Компоненты
 - **Manager**
 - отвечает за стратегию репликации
 - **Channel**
 - коммуникация между “нодами”



Репликация сессий

- SimpleTcpCluster
 - Компоненты
 - **Membership**
 - отвечает за контроль членства в кластере: отслеживает, когда ноды добавляются в кластер, либо наоборот его покидают



Репликация сессий

- SimpleTcpCluster
 - Компоненты
 - **Sender**
 - посылка сообщений членам кластера
 - **Receiver**
 - прием таковых



Репликация сессий

- SimpleTcpCluster
 - Компоненты
 - **Listeners**
 - Получают нотификации о событиях в кластере
 - **Deployer**
 - Farm-deployment



Репликация сессий

- SimpleTcpCluster
 - Компоненты
 - **Valves**
 - *Фильтры в обработке запросов от клиента*
 - **Interceptors**
 - **Valves** для сообщений между нодами



Репликация сессий

- SimpleTcpCluster
 - **DeltaManager**
 - All-to-All
 - Трафик между нодами квадратический: не рекомендуется для **большИх** кластеров
 - 100% failover



Репликация сессий

- SimpleTcpCluster
 - **DeltaManager**
 - не рекомендуется для ***большИх*** кластеров
 - документация предлагает разбивать кластер на группы маленьких кластеров



Репликация сессий

- SimpleTcpCluster
 - **BackupManager**
 - One-to-One-backup
 - Backup-нода не единая: каждая сессия имеет свою backup-ноду
 - Реплицируется “карта” кластера между всеми нодами
 - Трафик между нодами линейный



Репликация сессий

- SimpleTcpCluster
 - **BackupManager**
 - Как говорит документация:
 - not quite as battle tested as the delta manager.
 - Failover накладывает специфические ограничения



Репликация сессий

- SimpleTcpCluster
 - **BackupManager**

Если одна нода “закрешилась”:

- Если новая нода была бэкапной, то она становится основной и выбирается новая backup-нода



Репликация сессий

- SimpleTcpCluster
 - **BackupManager**

Если одна нода “закрешилась”:

- Если новая нода не содержала сессии, то она реплицирует сессию из бэкапной и становится основной



Репликация сессий

- **Стороннее решение**
 - Memcached
 - Hazelcast



Репликация сессий

- Стороннее решение
 - **Memcached**
 - Поставляется в виде привычного SessionManager'a (context.xml)
 - Аналогичен BackupManager'у
 - Обеспечивает failover memcached нод



Репликация сессий

- Стороннее решение
 - **Memcached**
 - Заявлена поддержка non-sticky сессий
 - Конфигурируемая стратегия сериализации
 - *Commercial support отсутствует*



Репликация сессий

- Стороннее решение
 - **Hazelcast Community**
- Поставляется в виде в виде ServletFilter'a
- Поддерживает 2 режима: P2P и Client/Server
- Аналогичен BackupManager'у



Репликация сессий

- Стороннее решение
 - **Hazelcast Community**
 - Поддерживает failover hazelcast нод
 - Заявлена поддержка non-sticky сессий
 - Конфигурируемая стратегия сериализации
 - Open Source



Репликация сессий

- Стороннее решение
 - **Hazelcast Enterprise**
- Поставляется в виде привычного SessionManager'a
- Поддерживает 2 режима: P2P и Client/Server
- Аналогичен BackupManager'у



Репликация сессий

- Стороннее решение
 - **Hazelcast Enterprise**
 - Поддерживает failover hazelcast нод
 - Заявлена поддержка non-sticky сессий
 - Конфигурируемая стратегия сериализации



Репликация сессий

- Стороннее решение
 - **Hazelcast Enterprise**
 - Open Source
 - Мониторинг из коробки
 - Commercial support 24/7
 - **ОЧЕНЬ ДОРОГО**



Вопросы & Ответы



Спасибо за внимание!