

# 我的一片云

应用JBoss AS 7构建企业云服务平台

张建锋

[jeff.zhang@jboss.org](mailto:jeff.zhang@jboss.org)

[weibo.com/findapple](http://weibo.com/findapple)

# 日程

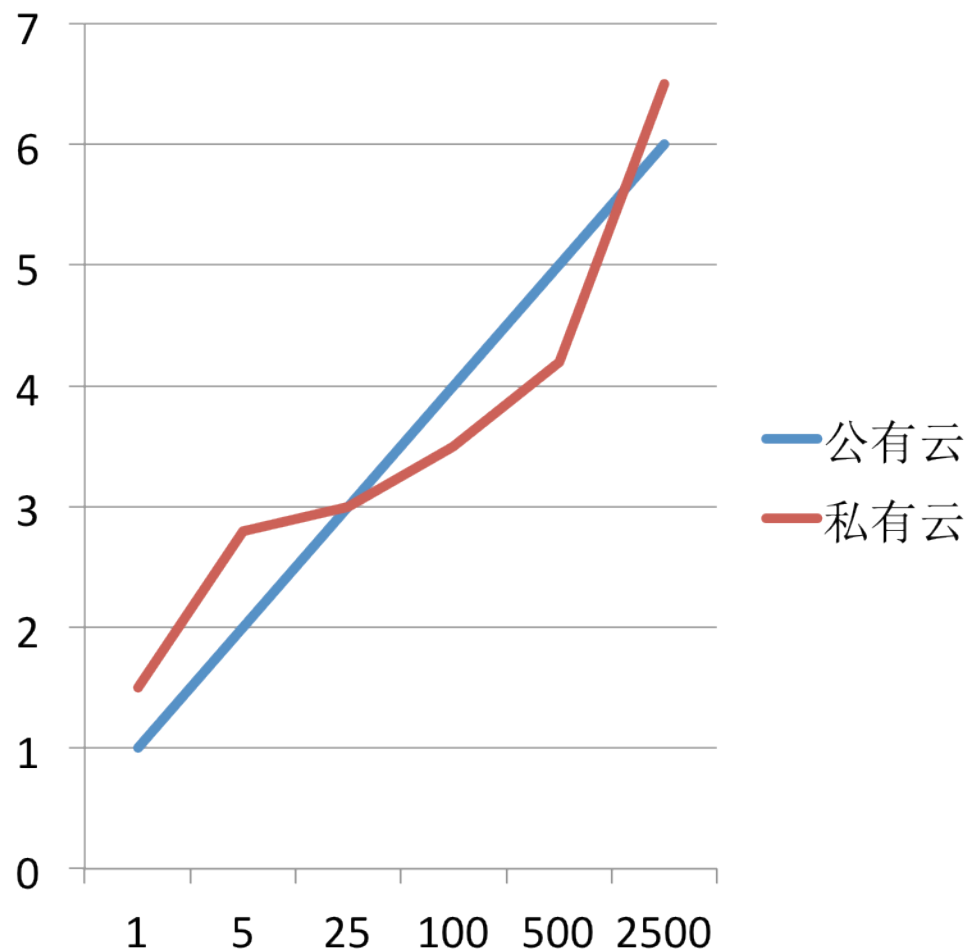
- 企业混合云
- 多种编程语言应用
- 与Paas平台融合
- 利用互联网云计算资源
- 多租户应用开发

# 当企业遇到云计算

- 云运算的本质是服务外包
  - IaaS, 购买机器, 搭建网络, 安装系统
  - PaaS, 配置服务, 系统调优, 基础组件
  - SaaS, 应用开发, 管理运维
- 节约成本, 快速开发部署
- 交给更专业的公司去做

# 需要自己的计算平台

- 信息完全掌控
- 数据安全考虑
- 开发运营有更多自主权
- 成本曲线(Y轴近似 $2^y$ , 指数坐标)



# 企业混合云

- 核心业务在私有云
- 网站，电子商务，社交化应用在公有云
- 突发流量等弹性需求，利用公有云
- 私有云计算能力和容量仔细规划，平稳扩展和更新
- (公有云中管理的虚拟机集群，是否也可以看成私有云计算平台?)

# 企业云计算平台需求

- 基本需求
  - 支持多种成熟应用
  - 支持业界主流开发技术方案
  - 方便部署，管理和运营
- 潜在需求
  - 方便的迁移应用到Paas上
  - 方便使用的互联网应用API
  - 多租户应用开发

# 日程

- 企业混合云
- 多种编程语言应用
- 与PaaS平台融合
- 利用互联网云计算资源
- 多租户应用开发

# Demo - Wordpress

- 知名博客程序
- PHP开发
- 利用quercus引擎



# 多语言应用服务器



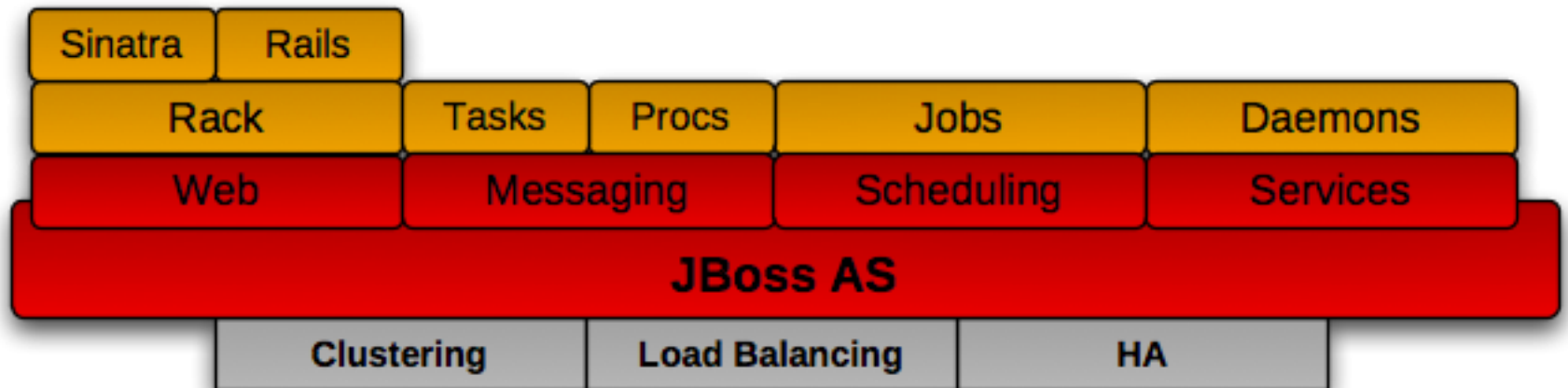
**TorqueBox**



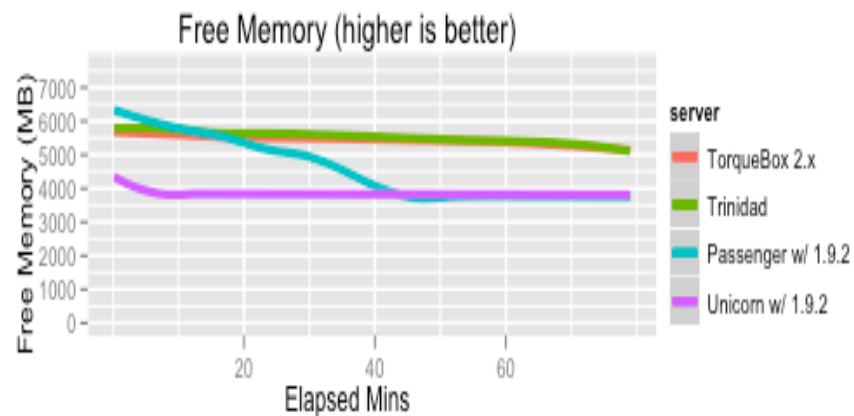
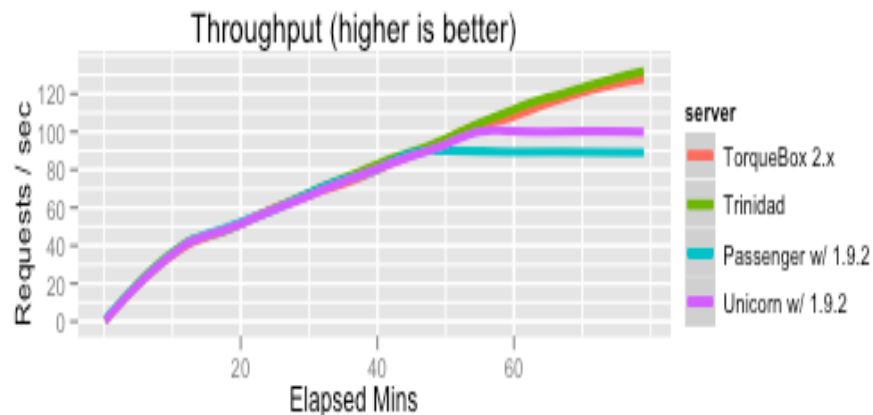
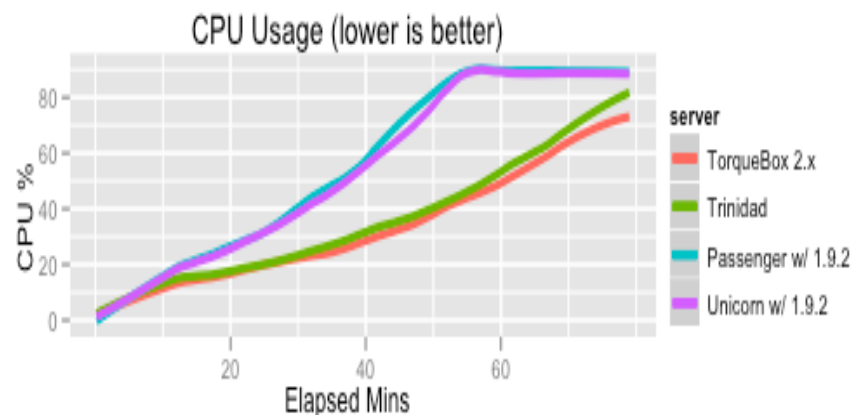
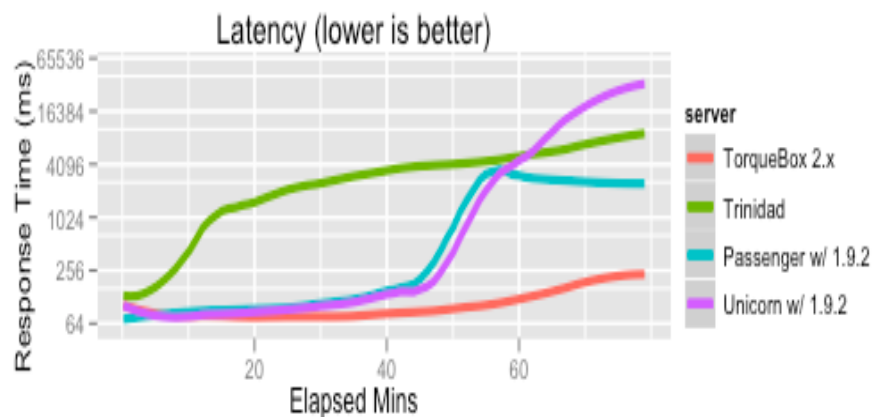
**Escalante**

# TorqueBox

- 基于JRuby, 支持Rails开发框架
- 利用JBoss AS 7应用服务器能力
  - 连接池
  - 集群
  - 消息系统



# 应用栈性能比较



# 多种语言平台应用

- 充分利用当前丰富的开源应用
  - 博客
  - 论坛
  - 电子商务
- Java平台，快速部署
- 动态语言
- 集成现有业务系统
- 性能优化

# 日程

- 企业混合云
- 多种编程语言应用
- 与**Paas**平台融合
- 利用互联网云计算资源
- 多租户应用开发

# Paas特点

- 可用的开发平台，已经配置并优化
- 只为使用的服务付费，不必考虑服务安装，网络调优等问题
- 互联网网站，移动APP服务器端，后台业务系统等都可以用
- 应用服务器发展的必然趋势



















# GAE特点

- Google Application Engine
- 优势
  - 成熟度高，已有海量应用
  - 技术实力强大，后台服务有保障
  - 多种主流语言开发支持
- 不足
  - 非标准Java，能力受到限制
  - 费用不菲
  - 国内访问受限

# CapeDwarf项目



- 类似SPI服务定义
- 模拟GAE开发环境

Appidentity API		Blobstore API		Capabilities API	
Channel API		Datastore API		Files API	
Images API		Logging API		Mail API	
Memcache API		Multitenancy API		OAuth API	
Prospective Search		Search API		Task Queue API	
URL Fetch API		Users API		XMPP API	



# 用本地服务代替

```
public class CapedwarfMemcacheServiceFactoryProvider extends CapedwarfFactoryProvider<IMemcacheServiceFactory> {  
    private final IMemcacheServiceFactory factory = new IMemcacheServiceFactory() {  
        public MemcacheService getMemcacheService(String s) {  
            return AspectFactory.createProxy(MemcacheService.class, new CapedwarfMemcacheService(s));  
        }  
  
        public AsyncMemcacheService getAsyncMemcacheService(String s) {  
            return AspectFactory.createProxy(AsyncMemcacheService.class, new CapedwarfAsyncMemcacheService(s));  
        }  
    }  
}
```

```
import org.infinispan.Cache;  
class CapedwarfMemcacheService implements MemcacheService {  
    protected final Cache<NamespacedKey, Object> cache;  
    public Object get(Object key) {  
        return cache.get(namespacedKey(key));  
    }  
}
```

# 应用服务器和Paas融合

- 应用所需要的服务基础设施是相似的
- 一次开发，多处部署
- 用户可以自由选择最合适的宿主环境
- 商业类比(各种配套环境)
  - 街边小店
  - 集贸市场
  - 购物中心

# Demo – blog4j

- 模拟SAE(SinaApp)开发平台
- 基于JBoss AS7

```
import com.sina.sae.memcached.SaeMemcache;
public class Cache {
    private static SaeMemcache memcache;
    static {
        try {
            memcache = new SaeMemcache();
        } catch (Exception e) {
            memcache = null;
        }
    }
    ...
}
```

# 日程

- 企业混合云
- 多种编程语言应用
- 与*PaaS*平台融合
- 利用互联网云计算资源
- 多租户应用开发

# 利用互联网资源

- 我们已经拥有大量的互联网资源
  - 存储，微博，博客，即时通信...
- 服务同质化
  - 相似的API(兼容性，标准)
  - 用户有选择权，比拼服务
  - 用户希望能迁移应用
- 和应用服务器诞生时的需求一致
  - 接口统一
  - 编程方式一致

# 典型云计算需求

- 用类似S3存储API，把重要的业务数据定期加密备份到国内云存储平台
- 企业重大产品发布，同时在有影响力的微博和信息通道发布消息
- 以上是中间件软件范围

# 通用云计算API

- jclouds

```
BlobStoreContext context = new BlobStoreContextFactory().createContext("aws-s3", identity, credential);
BlobMap map = context.createBlobMap("adrian.photos");

Blob blob = map.blobBuilder("sushi.jpg")
    .payload(new File("sushi.jpg"))// or byte[]. InputStream, etc.
    .contentDisposition("attachment; filename=sushi.jpg")
    .contentType("image/jpeg")
    .calculateMD5().build();

map.put(blob.getName(), blob);

context.close();
```

Provider	Maven Artifact ID
AWS	aws-s3
CloudOne	cloudestorage
HP	hpcloud-objectstorage
Microsoft	azureblob
Ninefold	ninefold-storage
Rackspace US	cloudfiles-us
Rackspace UK	cloudfiles-uk
Synaptic	synaptic-storage

# 日程

- 企业混合云
- 多种编程语言应用
- 与PaaS平台融合
- 利用互联网云计算资源
- 多租户应用开发



# 软件开发变革

- SaaS应用需求和PaaS宿主平台，使得多租户(Multi Tenancy)架构应用变得普遍
  - 博客平台
  - 淘宝店铺
- Web2.0引导用户创造内容，接下来每个用户都能提供商业服务
- 对软件开发商和最终用户都有影响

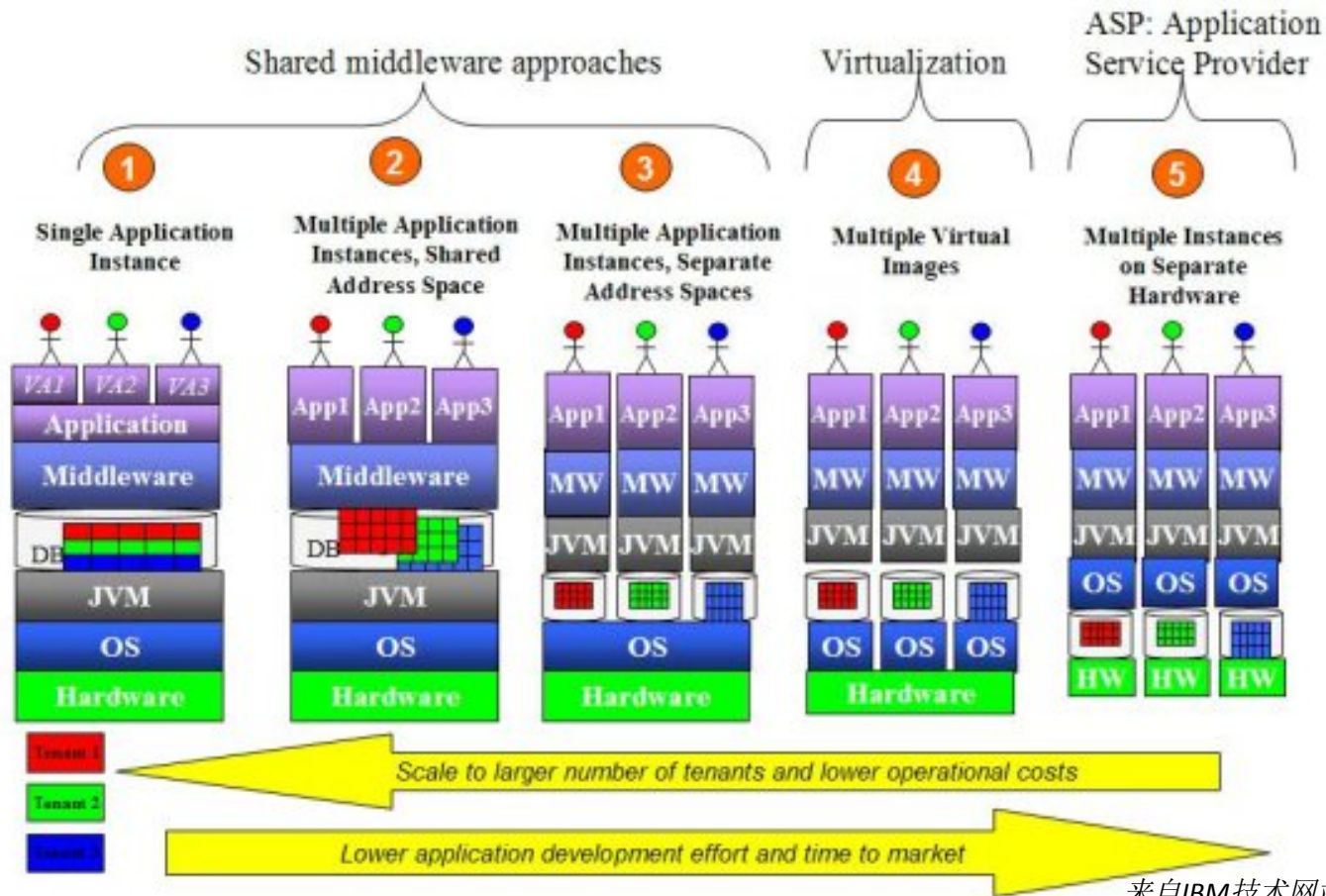
# 理想开发方式

- 软件开发商并不需要过多考虑架构变化
- 只需要符合标准接口就可以获得多租户能力
- 多租户的基础设施由应用服务器或Paas平台提供
- 大型Paas平台，可以聚合各个用户公开信息，提供统一门户入口，得到信息增值

# Java社区所做的努力

- 原计划JavaEE7中加入多租户的内容，但没有完成，推迟到JavaEE8
- 在一些局部规范已经有所体现
- 为了更好的提供云计算能力，一些改进的JRE也出现了
- JBoss组织积极推进云计算相关技术发展

# 多租户应用架构变化



来自IBM技术网站

# 应用设计要点

- 有状态的对象需要保存两个信息
  - 调用者
  - 应用名称(Id)
- 无状态的对象需要传递以上两个信息，需要的话进行逻辑处理
- GAE中多租户能力通过给以下服务提供namespace来区分
  - Persistent/Cache/Task

# JavaEE技术体系

- 应用可以通过URL区别
- Servlet定义了用户信息，需要传递应用Id
- SLSB传递应用Id，用户Id可通过Jaas获得
- JPA层需要应用Id来分离保存的数据
  - 不同数据库
  - 相同数据库和表，根据应用Id区分
- 未来应用Id留存在Context中，可以被不同应用层共享

# Hibernate对多租户的支持

- 获取Session,传入应用Id,按照一般方式开发

```
Session session = sessionFactory.withOptions()  
    .tenantIdentifier( yourTenantIdentifier )  
    ...  
    .openSession();
```

- 切换数据源

```
public class MultiTenantConnectionProviderImpl extends AbstractMultiTenantConnectionProvider {  
    private final ConnectionProvider acmeProvider = ConnectionProviderUtils.buildConnectionProvider( "acme" );  
    private final ConnectionProvider jbossProvider = ConnectionProviderUtils.buildConnectionProvider( "jboss" );  
    @Override  
    protected ConnectionProvider getAnyConnectionProvider() {  
        return acmeProvider;  
    }  
    @Override  
    protected ConnectionProvider selectConnectionProvider(String tenantIdentifier) {  
        if ( "acme".equals( tenantIdentifier ) ) {  
            return acmeProvider;  
        }  
        else if ( "jboss".equals( tenantIdentifier ) ) {  
            return jbossProvider;  
        }  
        throw new HibernateException( "Unknown tenant identifier" );  
    }  
}
```

# 常见多租户应用

- 数据进行分离
- 页面可以自行定制
  - 多套页面模版
- 逻辑基本相同，有一些简单的配置项
- 逻辑也可以很方便升级(灰度)，用户可以试用选择



# 开源应用服务器

- JBoss AS7
  - 先进的体系结构 MSC/JBossModule
  - 可拼装的子系统 Subsystem
  - 业界领先的JavaEE应用服务器
  - 完善的外延开源项目



谢谢大家！

[jboss.org/as7](http://jboss.org/as7)