

Works between the cloud



Davide Cerbo

Lavoro per Ex Machina in Lugano

(p.s.: stiamo selezionando gente sveglia)

<http://davide.cerbo.born-to-co.de>

http://twitter.com/davide_cerbo

<http://linkedin/in/davidecerbo>

davidecerbo@gmail.com



<http://exmachina.ch>



*Buongiorno,
oggi vi racconterò
una storia
(inventata)*



*Un giorno mi sveglio e mi viene una idea
per una fantastica applicazione:*

GA(E)STBOOK





- Apache Maven
- Spring Framework
- HSQL DB
- Apache OpenJPA

maven

Maven è una parola ebraica che vuol dire accumulatore di conoscenza.

Apache Maven è un sistema per gestire la fase di build di un progetto software.

Si occupa principalmente delle dipendenze, la fase di compilazione, di test e tramite un sistema di plugin anche di molto altro

La sua configurazione avviene tramite il file `pom.xml`.



Spring

E' un potente framework per lo sviluppo di software basato sul principio di Dependency Injection.

Nelle ultime versioni al classico file XML di configurazione ha aggiunto una serie di annotazioni



E' un database interamente scritto in Java.

Ha il vantaggio di essere molto leggero e può essere utilizzato direttamente all'interno dell'applicativo senza bisogno di servizi esterni.





Jetty è un HTTP server, un HTTP client, e un javax.servlet container.

Interamente scritto in Java ed integrato con Apache Maven tramite plugin.



The **Java Persistence API (JPA)** è uno standard che permette agli sviluppatori di gestire dati relazionali in modo indipendente dal database utilizzato.

Ne esistono diverse implementazioni, tra cui:
OpenJPA, Hibernate e DataNucleus

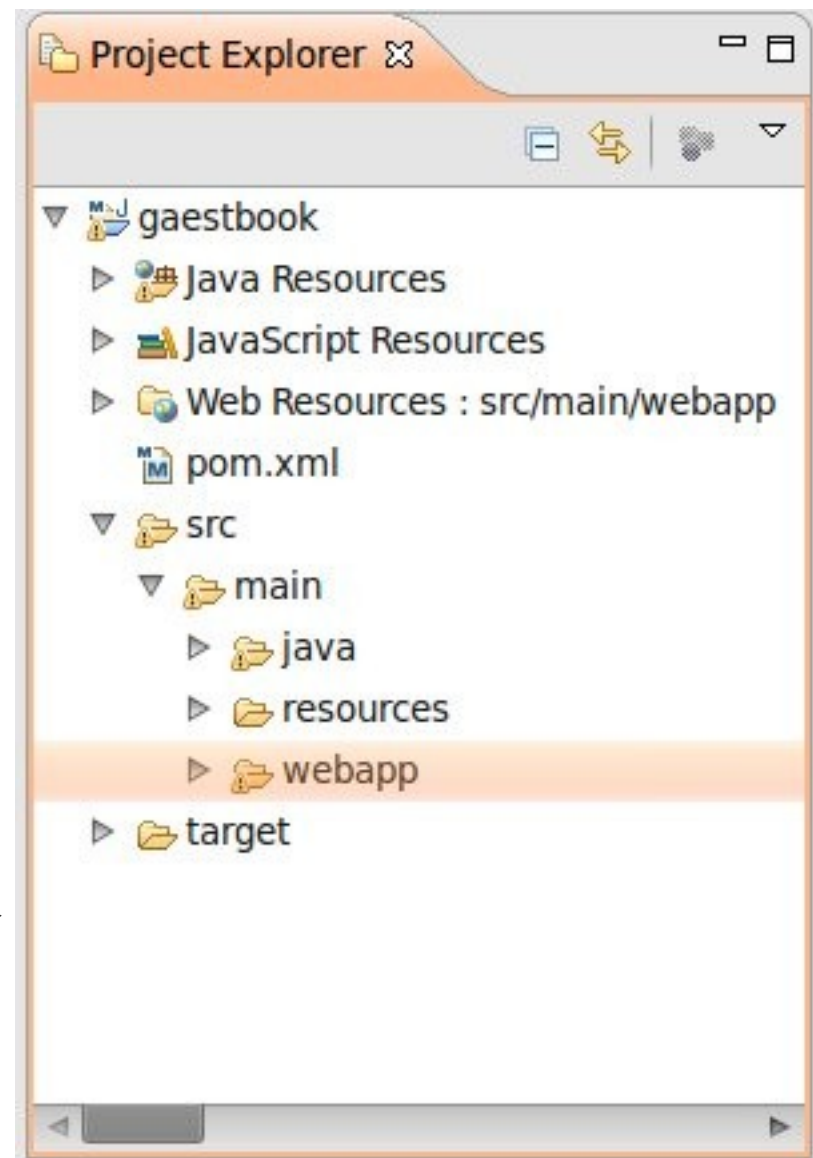


- Da shell lanciamo il comando:

```
mvn archetype:create -DgroupId=it.jesty  
-DartifactId=gaestbook -DarchetypeArtifactId=maven-  
archetype-webapp
```
- Creiamo la cartella java in /gaestbook/src/main
- Aggiungiamo il plugin di Jetty nel file pom.xml
- Aggiungiamo le dipendenze a: OpenJpa, HSQL, Spring WEBMVC, Spring ORM, etc...
- Digitiamo `mvn eclipse:eclipse` per generare i file di progetto per Eclipse


```
<project>

  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>3.0.0.RELEASE</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-orm</artifactId>
      <version>3.0.0.RELEASE</version>
    </dependency>
    <dependency>
      <groupId>org.apache.openjpa</groupId>
      <artifactId>openjpa-all</artifactId>
      <version>1.3.0-SNAPSHOT</version>
    </dependency>
    ...
  </dependencies>
  <build>
    <finalName>gaestbook</finalName>
    <plugins>
      <plugin>
        <groupId>org.mortbay.jetty</groupId>
        <artifactId>maven-jetty-plugin</artifactId>
        <version>6.1.10</version>
      </plugin>
      ...
    </plugins>
  </build>
</project>
```



Message.java

```
package it.jesty.gaestbook.bean;

import java.util.Date;

@Entity(name="Message")
public class Message {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column private String text;
    @Column private String email;
    @Column private String name;
    @Column private Date created;
    @Column private String uuid;

    public Message(String text, String email, String name, Date created) {
        this.text = text;
        this.email = email;
        this.name = name;
        this.created = created;
    }

    public Message() { /*NECESSARY!!!*/ }

    public Long getId() {
        return id;
    }

    public Date getCreated() {
        return created;
    }
}
```



GaestbookController.java

```
package it.jesty.gaestbook.controller;

import it.jesty.gaestbook.bean.Message;

@Controller
public class GaestbookController {

    private final MessageFacade messageFacade;

    @Autowired
    public GaestbookController(MessageFacade messageFacade) {
        this.messageFacade = messageFacade;
    }

    @RequestMapping(value="/messages", method= RequestMethod.GET)
    public ModelAndView showMessages(){
        ModelAndView modelAndView = new ModelAndView("messages-list");
        modelAndView.addObject("messages", this.messageFacade.listMessages());
        return modelAndView;
    }

    @RequestMapping(value="/messagePosted", method= RequestMethod.POST)
    public ModelAndView addMessages(@RequestParam(required=true) String name, @RequestParam String email, @RequestParam(value="message") String text){
        ModelAndView modelAndView = new ModelAndView("message-page");
        if(text.isEmpty() || email.isEmpty() || name.isEmpty()){
            modelAndView.addObject("error", "You must fill all fields to send a message!");
        } else {
            Message message = new Message(text, email, name, new Date());
            modelAndView.addObject("status", "Message sent successfully.");
            modelAndView.addObject("message", message);
            this.messageFacade.sendMessage(message);
        }
        return modelAndView;
    }
}
```



messages-list.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" isELIgnored="false" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<html>
  <head>
    <link rel="stylesheet" href="/styles/standard.css" type="text/css" />

    <title>G(A)estbook!</title>
  </head>
  <body>
    <h1>Welcome to G(a)estbook!</h1>
    <p>This is a sample web application developed with Spring Framework and OpenJPA...</p>
    <div id="messageFormContainer">
      <form action="/app/messagePosted" method="post">
        <div><label for="name">Name:</label></div>
        <div><input type="text" name="name" id="name"></input></div>
        <div><label for="email">E-mail:</label></div>
        <div><input type="text" name="email" id="email"></input></div>
        <div><label for="message">Message:</label></div>
        <div><textarea rows="5" cols="5" name="message" id="message"></textarea></div>
        <div class="buttonArea">
          <input type="submit" value="Post new message" />
          <input type="reset" value="Reset" />
        </div>
      </form>
    </div>
    <div id="messages">

      <c:forEach items="${messages}" var="item">
        <div class="message">
          <div class="message-list">
            <p>${item.text}</p>
          </div>
          <div class="footer">
            Posted by <span id="name-list">${item.name}</span> on <span id="email-list">${item.created}</span>
          </div>
        </div>
      </c:forEach>
    </div>
  </body>
</html>
```



message-page.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" isELIgnored="false" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<html>
  <head>
    <link rel="stylesheet" href="/styles/standard.css" type="text/css" />
    <title>G(a)estbook</title>
  </head>
  <body>
    <c:if test="${status != null}">
      <h2 id="message">${status}</h2>
    </c:if>

    <c:if test="${error != null}">
      <h2 id="error">${error}</h2>
    </c:if>

    <c:if test="${message != null}">
      <div class="message">
        <div class="message-list">
          <p>${message.text}</p>
        </div>
        <div class="footer">
          Posted by <span id="name-list">${message.name}</span> on <span id="email-list">${message.created}</span>
        </div>
      </div>
    </c:if>

    <div class="buttonArea">
      <a href="/app/messages"> &laquo; back</a>
    </div>
  </body>
</html>
```




```
application-context.xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema/context"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd
                           http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.0.xsd">

    <context:component-scan base-package="it.jesty.gaestbook" />

    <bean id="messageDao" class="it.jesty.gaestbook.dao.MessageJpaDao">
        <property name="entityManagerFactory" ref="entityManagerFactory"></property>
    </bean>

    <bean name="entityManagerFactory" id="entityManagerFactory"
          class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
        <property name="dataSource" ref="dataSource" />
        <property name="jpaVendorAdapter">
            <bean class="org.springframework.orm.jpa.vendor.OpenJpaVendorAdapter">
                <property name="showSql" value="true" />
                <property name="generateDdl" value="true" />
            </bean>
        </property>
        <property name="loadTimeWeaver">
            <bean class="org.springframework.instrument.classloading.SimpleLoadTimeWeaver" />
        </property>
    </bean>

    <bean id="dataSource"
          class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="driverClassName" value="org.hsqldb.jdbcDriver" />
        <property name="url" value="jdbc:hsqldb:file:javaday10" />
        <property name="username" value="sa" />
        <property name="password" value="" />
    </bean>

    <bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">
        <property name="entityManagerFactory" ref="entityManagerFactory" />
        <property name="dataSource" ref="dataSource" />
    </bean>

    <tx:annotation-driven />

</beans>
```


main-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema/context"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
    http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd
    http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.0.xsd">

  <context:component-scan base-package="it.jesty.gaestbook.controller" />

  <bean id="viewResolver"
    class="org.springframework.web.servlet.view.UrlBasedViewResolver">
    <property name="viewClass"
      value="org.springframework.web.servlet.view.JstlView" />
    <property name="prefix" value="/WEB-INF/jsp/" />
    <property name="suffix" value=".jsp" />
  </bean>

</beans>
```



persistence.xml ✕

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="1.0">  
  <persistence-unit name="gaestbook" transaction-type="RESOURCE_LOCAL"/>  
</persistence>
```



```
web.xml
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>

  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>WEB-INF/application-context.xml</param-value>
  </context-param>

  <filter>
    <filter-class>org.mortbay.servlet.WelcomeFilter</filter-class>
    <filter-name>WelcomeFilter</filter-name>
    <init-param>
      <param-name>welcome</param-name>
      <param-value>app/messages</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>WelcomeFilter</filter-name>
    <url-pattern>*</url-pattern>
  </filter-mapping>

  <listener>
    <listener-class>
      org.springframework.web.context.ContextLoaderListener
    </listener-class>
  </listener>

  <servlet>
    <servlet-name>main</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>main</servlet-name>
    <url-pattern>/app/*</url-pattern>
  </servlet-mapping>

</web-app>
```



`mvn jetty:run`



G(A)estbook!

http://localhost:8080/app/messages

Welcome to G(a)estbook!

This is a sample web application developed with Spring Framework and OpenJPA...

Name:

E-mail:

Message:

Post new message Reset

Benvenuti!

Posted by Davide on Fri Jan 29 00:38:13 CET 2010

Non so perché, ma...



...milioni di utenti usano G(a)estbook!



Global

By Country

By Category



Top Sites

The top 500 sites on the web. ?

1. G(a)estbook

<http://davide.cerbo.bom-to-co.de>

A simple guestbook.



2. Google

google.com

Enables users to search the Web and images. Features include PageRank, caching and translation of results, and an option to view similar pages. The company's focus is developing search technology.

3. Facebook

facebook.com

A social utility that connects people, to keep up with friends, upload photos, share links and videos.

4. YouTube - Broadcast yourself

youtube.com

YouTube is a way to get your videos to the people who matter to you. Upload, tag and share your videos worldwide!

5. Yahoo!

yahoo.com

Personalized content and search options. Chatrooms, free e-mail, clubs, and pager.

6. Windows Live

live.com

Search engine from Microsoft.

7. Wikipedia

wikipedia.org

An online collaborative encyclopedia.

8. Blogger.com

blogger.com



Free Download

Top 1,000,000 Sites (Updated Daily)

Ads by Google

Google Website Optimizer

Testen Sie, welche Landing Page die meisten Conversions erzielt. Gratis

www.google.com/WebsiteOptimizer

Professional Network Site

Build your profile and make great Professional Connections at Vindex!

www.Vindex.com

Search Engine Submission

Add our button to get more people to your website or blog. Add free.

www.TheBest.info.com

Nettunes Windows XP

Reparatör des Errors, Augmentiert in Videos, Tellerchanger maintenance...

RegistrierenWindows.com

Ads by Google

Google Website Optimizer

Testen Sie, welche Landing Page die meisten Conversions erzielt. Gratis

www.google.com/WebsiteOptimizer

Nettunes Windows XP

Reparatör des Errors, Augmentiert in Videos, Tellerchanger maintenance...

RegistrierenWindows.com

Search Engine Submission

Add our button to get more people to your website or blog. Add free.

www.TheBest.info.com



HTTP Error 408 Request timeout

e non solo...





Devo migliorare le prestazioni del mio applicativo.
Posso scegliere se:

- **Scalare verticalmente:** aggiungere potenza di calcolo al singolo server.
- **Scalare orizzontalmente:** aggiungere server che collaborano tra di loro

- Prima o poi raggiungerà la massima configurazione possibile
- Non ho abbastanza soldi per acquistare l'ultimo server presente sul mercato



- Non sono un sistemista
- I server hanno un prezzo elevato
- L'housing ha prezzi elevati
- Costruire e gestire una web farm privata per pochi server ha un prezzo elevato
- Se il sito non cresce come mi aspetto ho sprecato tempo e denaro

Cerco su Google e vedo che alcune aziende offrono servizi di Cloud o Grid Computing

...uhm...





your servers are not bulletproof
WATCH NOW

your servers will blow up on you
WATCH NOW

your servers will burn!!!
WATCH NOW

NO HARDWARE dot com

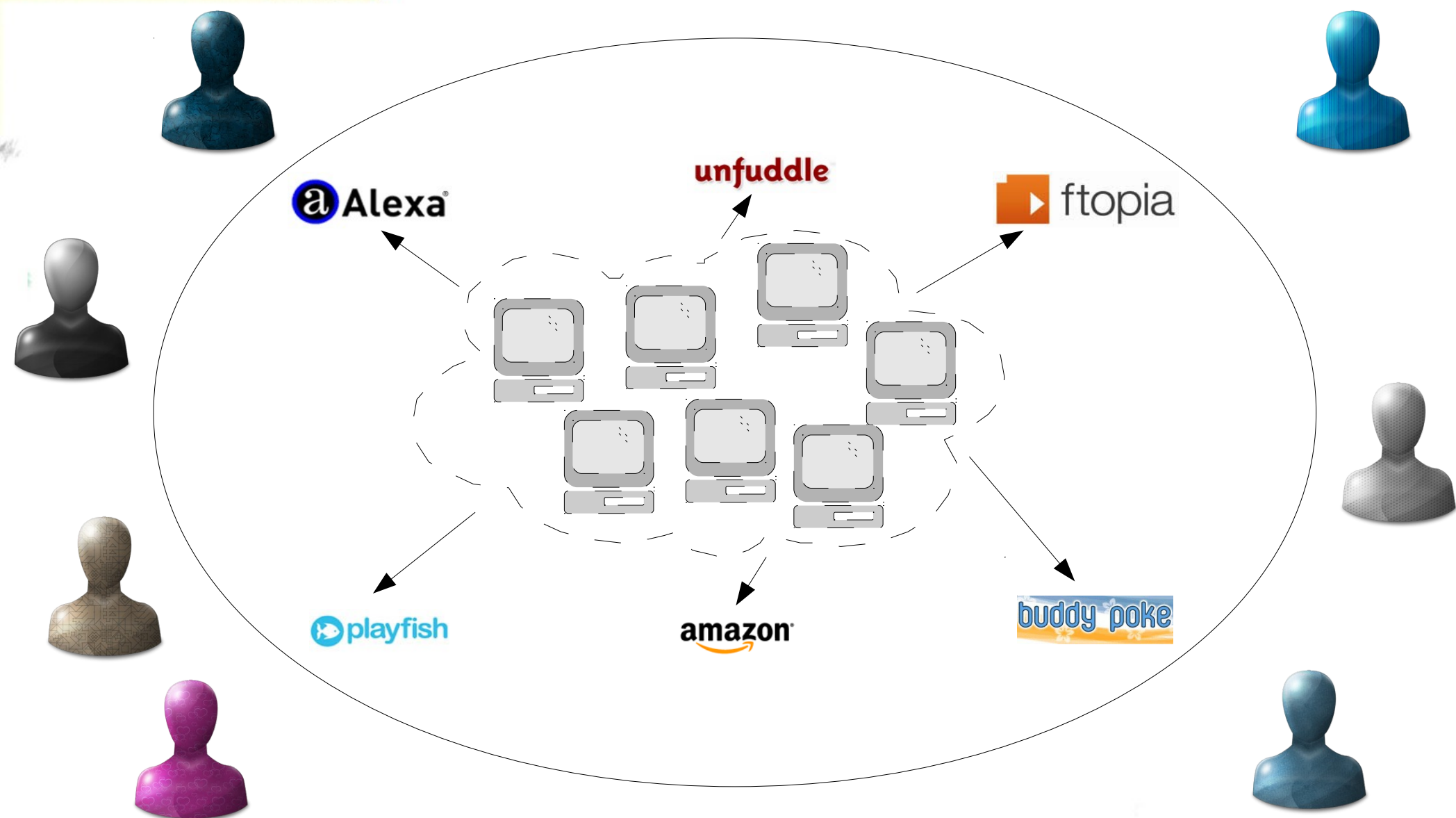
POWERED BY **GO GRID**





- Molte richieste
- Piccoli compiti
- Poche richieste
- Grandi lavori

Molti **SERVER** lavorano
insieme per **TE**



JAVADAY^{IV} Perché un servizio di cloud computing?

- Non siamo amministratori di sistema o tecnici hardware
- Niente ravvio dei server, niente problemi di rete
- Non abbiamo molti server a nostri disposizione
- Non abbiamo i soldi per acquistarli



Google Datacenter
a Dallas, Oregon

Picture credit: Portland Java

User Group 18 Agosto 2009



- Quando la nuvola non può fornire una risorsa per la richiesta
- Quando è necessario l'intervento umano per creare una nuova risorsa per la nuvola



<http://www.gridgain.com>



appscale

Open Source Platform for Google AppEngine Apps

<http://code.google.com/p/appscale/>



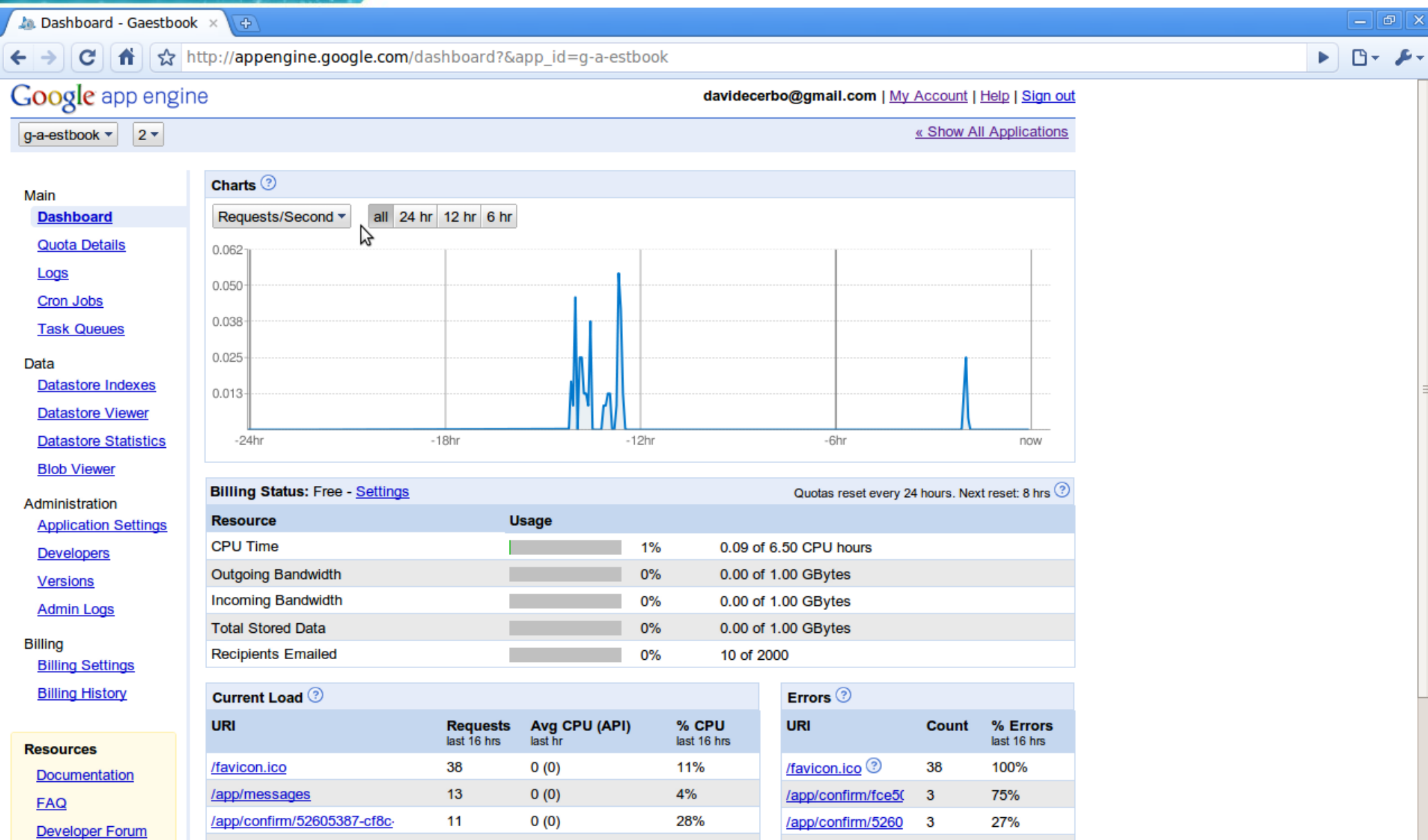
**Eucalyptus
Systems**

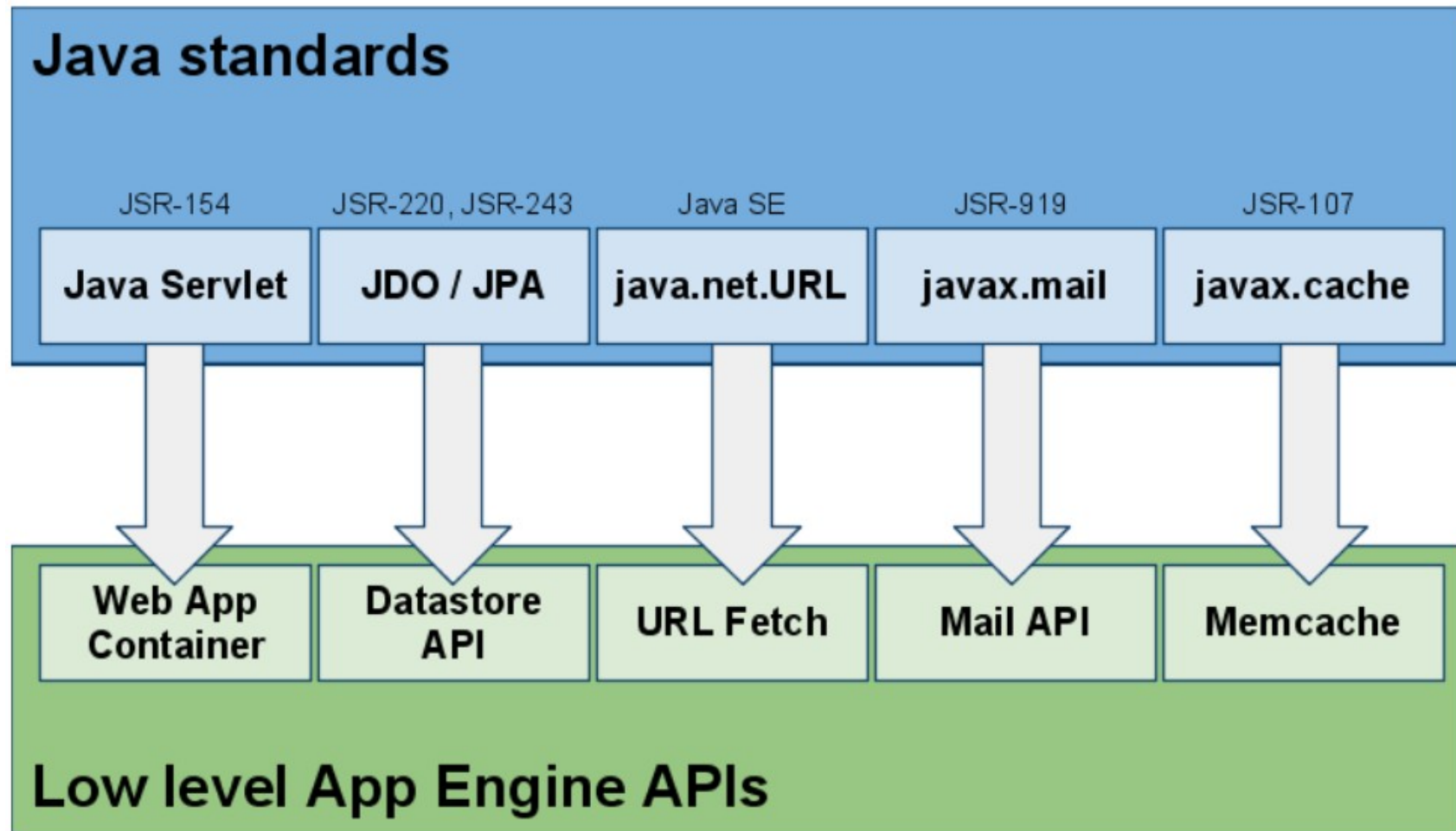
<http://www.eucalyptus.com>

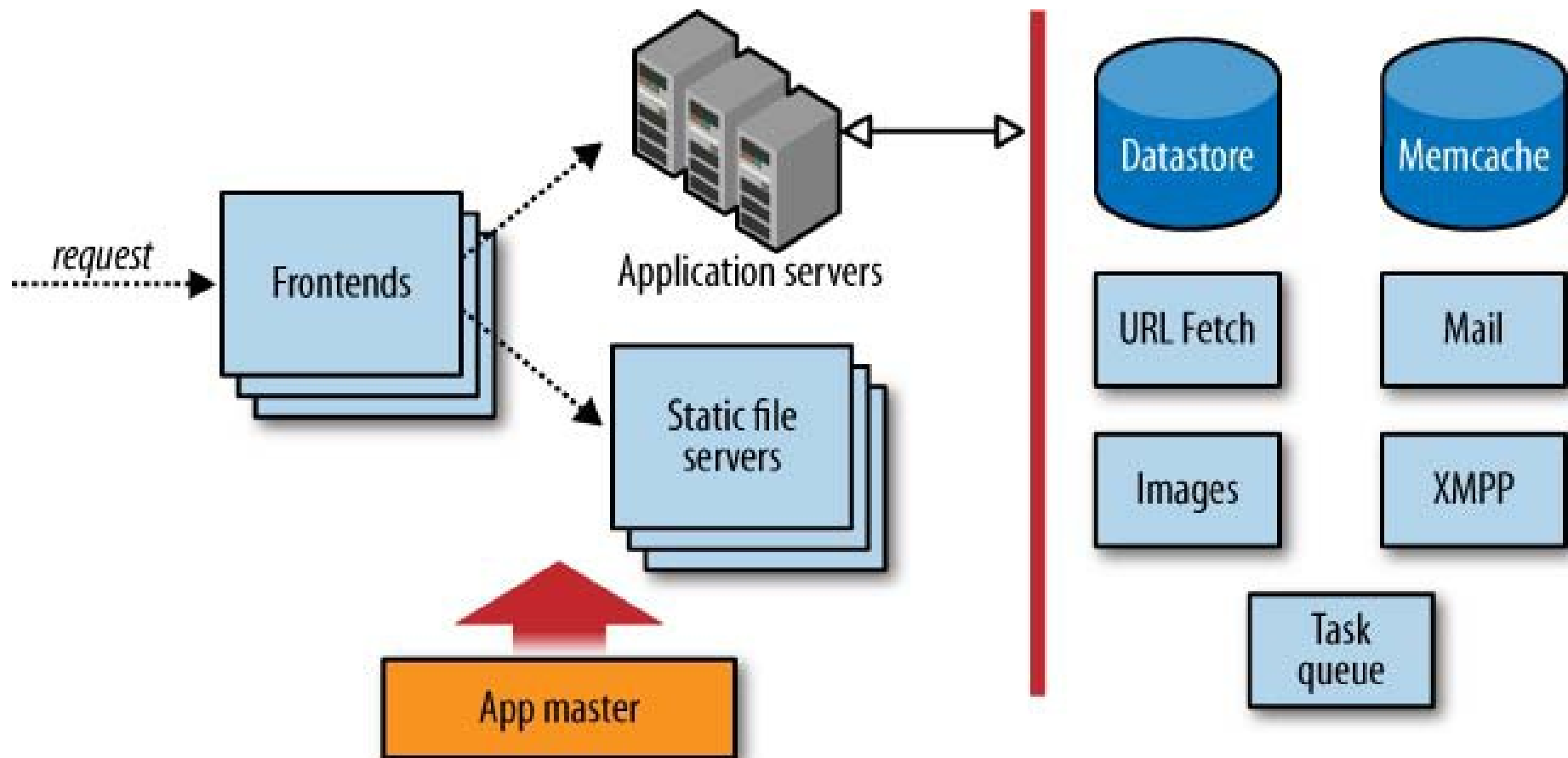




- L'amministrazione di sistema è a prova di stupido
- Economico, paghi solo se l'applicazione cresce
- Ottimo software installati come Memcache, Bigtable, etc.
- Una ottima console di amministrazione con log di sistema, statistiche di accesso, statistiche di consumo, gestione dell'applicazione
- Compatibile con **JAVA**, PYTHON e altri linguaggi basati sulla JVM: Scala, Groovy, etc...
- Compatibile con molti standard Java
- Non c'è sistema operativo, un pensiero in meno
- Molte limitazioni, ma queste ci porteranno ad ottenere a **scalare e a bilanciare il carico in automatico**







- **Memcache API** – cache key-value in memoria ad alte prestazioni
- **Datastore** – database
- **URLFetch** – invocazione di url esterni
- **Mail** – sistema invio e-mail
- **Task Queues** – invocazioni in background. Ottima al posto dei Thread.
- **Images** – per la manipolazione delle immagini
- **XMPP** – protocollo di comunicazione real-time usato anche da Google Talk
- **Cron Jobs** – schedulazione di task temporizzata
- **User Accounts** – autenticazione tramite gli account di Google

<http://code.google.com/appengine/docs/java/apis.html>



- HTTP request devono essere eseguite in 30 secondi, altrimenti GAE lancia una `DeadlineExceededException`
- La prima request potrebbe essere lenta
- No socket
- No threads
- Solo 1000 file per applicazione



- 1,300,000 richieste http
- In/Out Bandwidth 1 Gb
- CPU Time 6.5 CPU-ora
- Dati memorizzati 1 Gb
- Mail inviate 7,000
- Mail verso tutti 2,000
- Mail verso l'amministratore 5,000
- Url fetch 657,000
- Chiamate XMPP 657,000
- Manipolazione immagini 864,000
- Chiamate verso Memcache 8,600,000
- Task 100,000
- E molto altro...

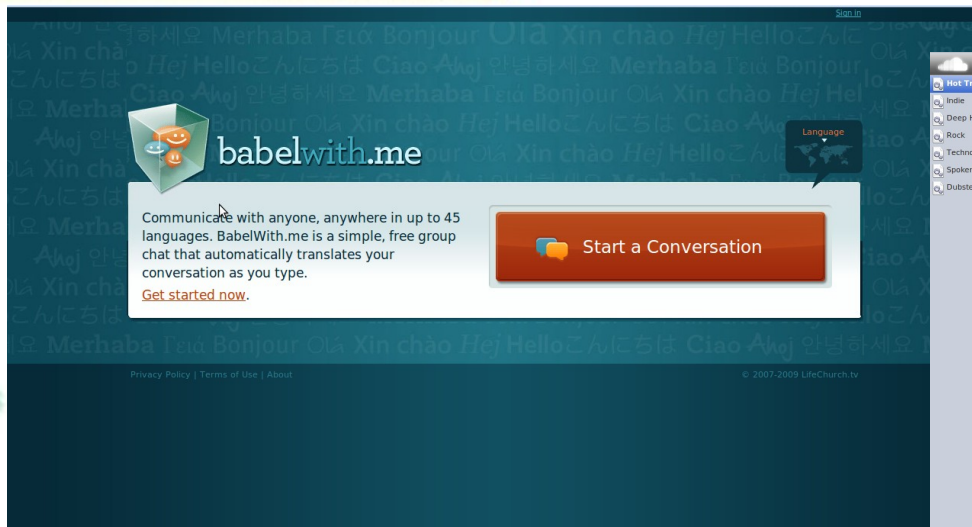


- Outgoing Bandwidth.....\$0.12 / Gb
- Incoming Bandwidth.....\$0.10 / Gb
- CPU Time.....\$0.10 / h
- Stored Data.....\$0.15 / Gb per mese
- Recipients Emailed.....\$0.0001 / destinatari

(Tasse escluse)

Potete scegliere il vostro budget massimo giornaliero per evitare brutte sorprese!





babelwith.me

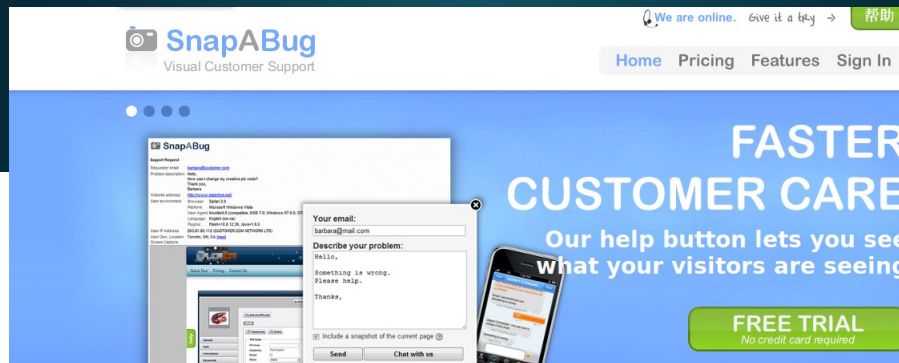
Communicate with anyone, anywhere in up to 45 languages. BabelWith.me is a simple, free group chat that automatically translates your conversation as you type.

[Get started now.](#)

[Start a Conversation](#)

Privacy Policy | Terms of Use | About

© 2007-2009 LifeChurch.tv



SnapABug
Visual Customer Support

[Home](#) [Pricing](#) [Features](#) [Sign In](#)

FASTER CUSTOMER CARE

Our help button lets you see what your visitors are seeing

FREE TRIAL
No credit card required

Your email:

Describe your problem:

Email

Name

Password

By clicking Start your SnapABug Trial, you agree to the [Terms of Service](#) and [Privacy Policy](#).

Some of our customers:

SENDGRID webintellix                                 

Integrations:

With SnapABug your support requests go directly to your preferred CRM, Help Desk or Bug Tracking solution. You can also simply receive them by email.

 **salesforce.com**
Success. Not Software.

 **zendesk**
LOVE YOUR HELP DESK

 **PIVOTALTRACKER**

 **FogBugz**



Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>



THE CLOUD PLAYER

The Cloud Player is a web-based music player that lets you...

- Find and play all tracks from SoundCloud
- Save playlists to your Google Account
- Make smart playlists based on genre, BPM, etc
- Share your ready-made playlists with your friends
- Collaborate in making the best playlists ever

Build by Henrik Berggren & Eric Wahlforss using Google App Engine, Python, jQuery, SoundManager 2, Crystal and the SoundCloud API. Also thanks to jsm for the back-button support. Do you want to contribute? Go ahead and fork from GitHub!

Title	Artist	Time	Description	BPM	Genre
Ball	NIT GRIT	4:57		137	Dubstep/Glitch...
Colouring of Pigeons	The Knife	11:01	The Knife, in collaboration with Mt. Sim...		
Ugmo	Shatterfreak	4:33	This is what happens when two ugly b...	170	Ugly Ugly Dub...
The Touch	ASC	2:30	Fortcoming on Autonomic		Autonomic
MODERAT - A New Error	BPitch Control	6:07	taken from the album "Moderat" (BPC2...		
Squits and Giggles	NIT GRIT	5:17	This track was made for my good frien...		Dubstep/Heav...
Technical Touch	NIT GRIT	4:33		145	Dubstep/Heav...
Helios - First Dream Called Ocean (Stra...	straydnb	4:43		170	Dubstep/Heav...
Yeah Yeah Yeahs - Maps (Elite Force Re...	Elite Force	3:32	It's a bigroom electro-mashup, pure & ...		Mashup
Heartbeats	The Knife	3:51			
Silent Shout	The Knife	4:53			
Kinda Andromeda (Mechanistic Andromeda)	mechano	3:30	Remix of Handful Andromeda from Mech...		Bass, Glitcho...
um & Bass					um & Bass
ico funk					ico funk
istatep					istatep
Beat - Hou...					Beat - Hou...
imal Techno					imal Techno
um & Bass					um & Bass
um & Bass					um & Bass
h House					h House
chno House					chno House
um and bass					um and bass
edit					edit
um & Bass					um & Bass
etroHouse					etroHouse
ie it slow, r...					ie it slow, r...
um and Bass					um and Bass
ic at the ...					ic at the ...
Eric Sumo - The Real Moustache (Fine ...	Fine Cut Bodies	5:32	My remix for one of my fav producers: ...		Glitchhop
Anile - Induction	ANILE	4:09	stripped back rollin bass line like it sho...		drum n bass
Spheric Universe 'extended mix REILO...	Cookieazz	6:59	TAKE2 - EDITED...WAT was i thinking? ...	127	Synthetic Sph...
Toto - Africa (AC Re-Edit)	Andrew Clarke	7:08		95	80's
Naibu - Use of Weapons (Warm Comm...	Warm Communicat...	2:27	Warm Communications 2010 "Embers...		
Tina Turner-What's Love Got To Do Wit...	Lord of the Idles	8:13	remix: Happy New Year to you all and ...	104	
Marcus Schmidt vs Double C - es schall...	Marcus Schmidt D...	7:21	VA. Prefix EP @ Farbon Records		



Davide Cerbo - davidecerbo@gmail.com – Ex Machina

Javaday IV – Roma – 30 gennaio 2010



- Creiamo un nuovo progetto usando Maven-Gae-Plugin:

```
mvn archetype:create -DarchetypeGroupId=net.kindleit  
-DarchetypeArtifactId=gae-archetype-gwt  
-DarchetypeVersion=0.5.0 -DgroupId=your.groupId  
-DartifactId=your-artifactId  
-DremoteRepositories=http://maven-gae-  
plugin.googlecode.com/svn/repository
```

- Facciamo un po' di pulizia e copiamo il sorgente dal precedente progetto

Oppure:

- Seguiamo questa guida per migrare il pom.xml:
http://www.kindleit.net/maven_gae_plugin/usage.html

application-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema/context"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd
                           http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.0.xsd">

    <context:component-scan base-package="it.jesty.gaestbook" />

    <tx:annotation-driven />

    <bean id="messageDao" class="it.jesty.gaestbook.dao.MessageJpaDao">
        <property name="entityManagerFactory" ref="entityManagerFactory"></property>
    </bean>

    <bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalEntityManagerFactoryBean">
        <property name="persistenceUnitName" value="transactions-optional" />
    </bean>

    <bean name="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">
        <property name="entityManagerFactory" ref="entityManagerFactory" />
    </bean>

</beans>
```

La configurazione del
Datasource é nel
persistence.xml



persistence.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd"
  version="1.0">
  <persistence-unit name="transactions-optional">
    <provider>org.datanucleus.store.appengine.jpa.DatastorePersistenceProvider</provider>
    <properties>
      <property name="datanucleus.NontransactionalRead" value="true"/>
      <property name="datanucleus.NontransactionalWrite" value="true"/>
      <property name="datanucleus.ConnectionURL" value="appengine"/>
    </properties>
  </persistence-unit>
</persistence>
```



web.xml

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
```

```
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>WEB-INF/application-context.xml</param-value>
  </context-param>
```

```
  <listener>
    <listener-class>
      org.springframework.web.context.ContextLoaderListener
    </listener-class>
  </listener>
```

```
  <servlet>
    <servlet-name>main</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
```

```
  <servlet-mapping>
    <servlet-name>main</servlet-name>
    <url-pattern>/app/*</url-pattern>
  </servlet-mapping>
```

Per utilizzare collezioni e oggetti
con lazy loading abilitato

```
  <filter>
    <filter-name>OpenEntityManagerInViewFilter</filter-name>
    <filter-class>org.springframework.orm.jpa.support.OpenEntityManagerInViewFilter</filter-class>
  </filter>
```

```
  <filter-mapping>
    <filter-name>OpenEntityManagerInViewFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
```

```
  <welcome-file-list>
    <welcome-file>app/messages</welcome-file>
  </welcome-file-list>
```

```
</web-app>
```



```
appengine-web.xml
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application>g-a-estbook</application>
  <version>1</version>

  <!-- Configure java.util.logging -->
  <system-properties>
    <property name="java.util.logging.config.file" value="WEB-INF/classes/logging.properties"/>
  </system-properties>
</appengine-web-app>
```

GaeEntityManagerFactory.java

```
package it.jesty.gaestbook.gae;

import java.util.Map;

public class GaeEntityManagerFactory implements EntityManagerFactory{

    private static final EntityManagerFactory emfInstance = Persistence.createEntityManagerFactory("transactions-optional");

    public GaeEntityManagerFactory() {}

    public EntityManager entityManager() {
        return emfInstance.createEntityManager();
    }

    @Override
    public void close() {
        emfInstance.close();
    }

    @Override
    public EntityManager createEntityManager() {
        return emfInstance.createEntityManager();
    }

    @Override
    public EntityManager createEntityManager(Map map) {
        return emfInstance.createEntityManager(map);
    }

    @Override
    public boolean isOpen() {
        return emfInstance.isOpen();
    }
}
```



EmailServiceImpl.java

```
package it.jesty.gaestbook.service;

import java.util.Properties;

@Component
public class EmailServiceImpl implements EmailService {

    private static final Logger log = Logger.getLogger(EmailServiceImpl.class.getName());

    @Override
    public void sendMail(String from, String to, String subject, String text) {
        Properties props = new Properties();
        Session session = Session.getDefaultInstance(props, null);

        try {

            Message msg = new MimeMessage(session);
            msg.setFrom(new InternetAddress(from));
            msg.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
            msg.setSubject(subject);
            msg.setText(text);
            Transport.send(msg);

        } catch (Exception e) {
            log.severe("Error sending mail to " + to + ". Error: " + e.getMessage());
        }
    }
}
```



MessageFacadeImpl.java

```
public void sendMessage(Message message){
    String uuid = UUID.randomUUID().toString();
    message.setUuid(uuid);
    this.messageDao.createMessage(message);
    this.emailService.sendMail("davidecerbo@gmail.com", message.getEmail(), "G(a)estbook", buildText(message));
}

public Message confirmMessage(String uuid){
    return this.messageDao.confirmMessage(uuid);
}

public Collection<Message> listMessages(){
    return this.messageDao.list();
}

private String buildText(Message message) {
    return new StringBuilder("Hi, ")
        .append(message.getName())
        .append(" you have posted this message: \n")
        .append("\n")
        .append(message.getText())
        .append("\n\n")
        .append("Please confirm it visiting this url: ")
        .append("http://g-a-estbook.appspot.com/app/confirm/")
        .append(message.getUuid())
        .append("\nBest regards,\n G(a)estbook")
        .toString();
}
```



<http://code.google.com/p/gae-javaday10/>

```
svn checkout http://gae-  
javaday10.googlecode.com/svn/trunk/ gae-javaday10-  
read-only
```



- Minimizzare il traffico dati tramite una architettura 2.5 tier
- Le operazioni che non danno un responso immediato all'utente vanno eseguite nei task (*vedi invio e-mail*)
- Inserire in cache tutte le informazioni più recenti (*lista degli ultimi messaggi da visualizzare nella home*)
- Eseguire operazioni batch che possono alleggerire il sistema tramite i task di tipo cron (*cancellare i messaggi non confermati entro tre giorni*)

- Spring Framework
<http://www.springsource.org>
- Apache Maven
<http://maven.apache.org>
- Apache OpenJPA
<http://openjpa.apache.org>
- Hsql DB
<http://hsqldb.org>
- Jetty
<http://jetty.codehaus.org>
- Google App Engine
<http://appspot.com>
- Maven GAE Plugin
<http://code.google.com/p/maven-gae-plugin/>
- Datanucleus
<http://www.datanucleus.org>



- How Buddypoke scales on Facebook using GAE
<http://highscalability.com/blog/2010/1/22/how-buddypoke-scales-on-facebook-using-google-app-engine.html>
- Designing application for cloud deployment
<http://gojko.net/2010/01/25/designing-applications-for-cloud-deployment/>
- Google App Engine for Java - 3 Tips for Getting Started
<http://www.pardontheinformation.com/2009/06/google-app-engine-for-java-3-tips-for.html>
- The Softer Side Of Schemas - Mapping Java Persistence Standards To the Google App Engine Datastore
<http://code.google.com/events/io/2009/sessions/SofterSideofSchemas.html>
- Cloud versus cloud
<http://www.infoworld.com/d/cloud-computing/cloud-versus-cloud-guided-tour-amazon-google-appnexus-and-gogrid-122>



