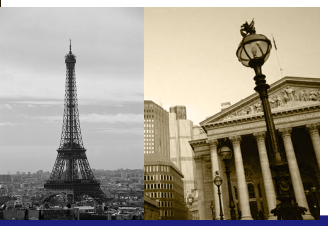


Nov 2009

Technologies Java pour le Web

Principes, Mise en Oeuvre

Arnaud NAUWYNCK
arnaud.nauwynck@gmail.com

Plan

- Introduction, Html
- Protocol Http
- Contenu Dynamique coté server
  - Cgi-bin => Java Servlets, Jsp
  - Apache, Tomcat, Architecture webs
  - Frameworks MVC
- Contenu Dynamique coté client
  - JavaScript
  - GWT : Google Web Toolkit

Introduction

- Début d'Internet (ip, tcp)
  - origine : universitaire / militaire
  - but: connecter des pcs
- Début du Web (http, html)
  - origine : Cern (labo de recherche)
  - but: documentation, liens hypertextes...
- Dévelopements exponentiels des standards, extensions propriétaires, librairies dev
  - Ex: webservices soap, protocoles over http

Format Html

- Html : Hyper-Text Markup Langage
- Re-normaliser en xhtml : xml (balises fermées)
- Exemple:

```
<?xml>
<head> … </head>
<body>
   <h1>Titre</h1>
   <p>Un paragraph de text</p>
   <A href=”http://host/a.html”>un lien</A>
</body>
```

Html

- Html en apparence pauvre
  comparé à latex, pdf, ps, MsWord...
- Mais intégration avec
  - des liens servers (contenus dynamiques)
  - des images (statiques, gif animées)
  - des applets (java, activex, flash...)
  - des scripts (javascript, ajax, dom, json…)
  - des feuilles de style (css, xsl)
- … finalement très riche et complexe

Protocol HTTP

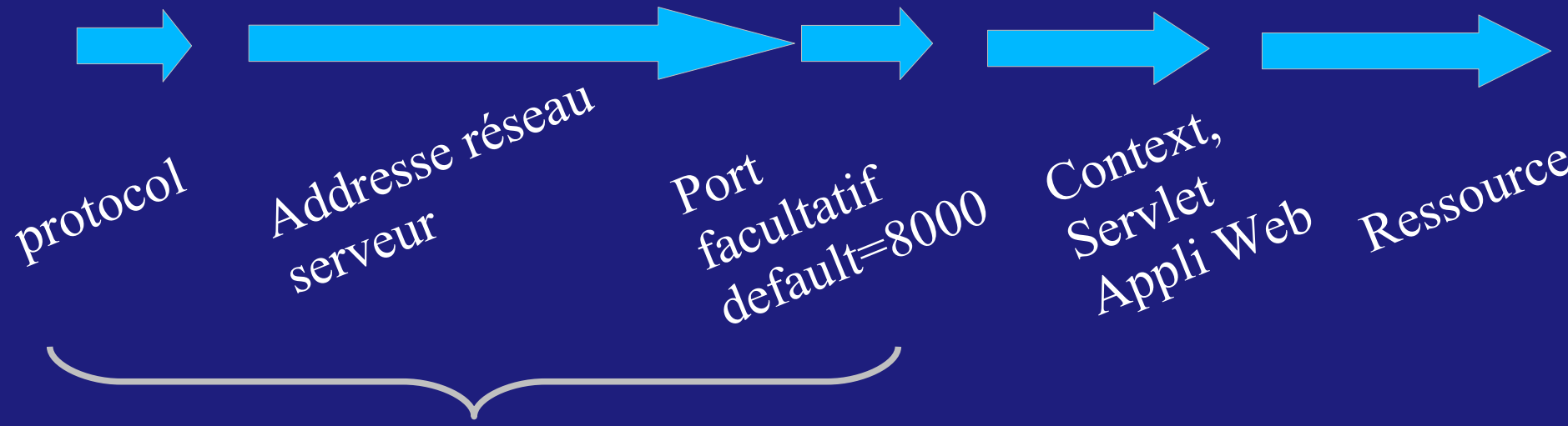
- HTTP = Hyper-Text Transfer Protocol
- Protocol client-server, en texte clair
- En général au dessus d'une socket TCP... niveau 7 dans la couche réseau
- Le Server écoute par convention sur le port 80, 8000 ou 8080

- Le server attend les requètes des clients Pour chaque requète, il répond

SUNGARD

Liens URL – Socket - HTTP

- http://hostname.domaine:8080/webapp/login.jsp



Traité par Client (Browser)
… pour connection TCP

Traité par Server
(Container Web) …
redirige vers context

Interprété pour
Context Appli

Lien Http - Socket - Browser

- Le browser Web cache tout le travail:
  - Parse URL
  - Lookup DNS + connect
  - Send HTTP GET/POST + parameters
  - Send Cookies
  - Read results (header: cookies) + text
    - Parse result text
    - Repeat call server … for Icons, sub frames
  - Store Cookies

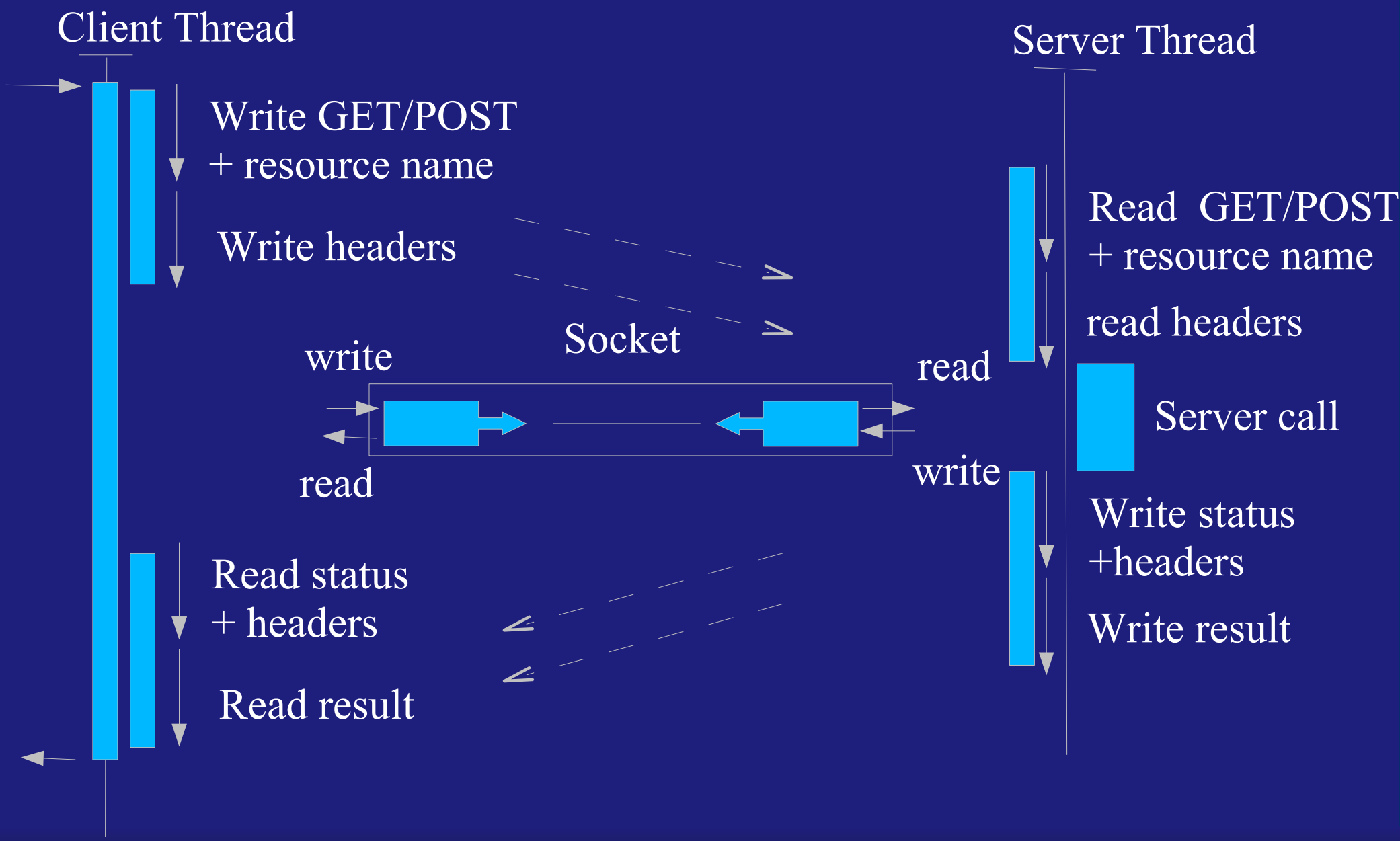
SUNGARD

Déroulement d'un appel côté client

- Le client se connecte, choisit une action:
  HTTP GET, POST, PUT, REMOVE
- fournit
  - nom de la resource
  - des arguments, des headers (cookies)
- .... Une ligne blanche … et attend
- lit la réponse
  - Code 401..404, des headers
  - Le texte
- .. et se déconnecte

SUNGARD

Déroulement d'un Appel



Déroulement d'un Appel coté serveur

- Le serveur attend des connections
  serverSock = ServerSocket.listen()
- Pour chaque connection,
  clientConn = serverSock.accept();
- Il lit les inputs
  clientConn.read().... until blank line
- Et répond
  clientConn.write();
- Puis referme la socket
  clientConn.close();

Utilisations de HTTP

- En général, pour récupérer des contenus de pages statiques
  - des fichiers html
  - des icons

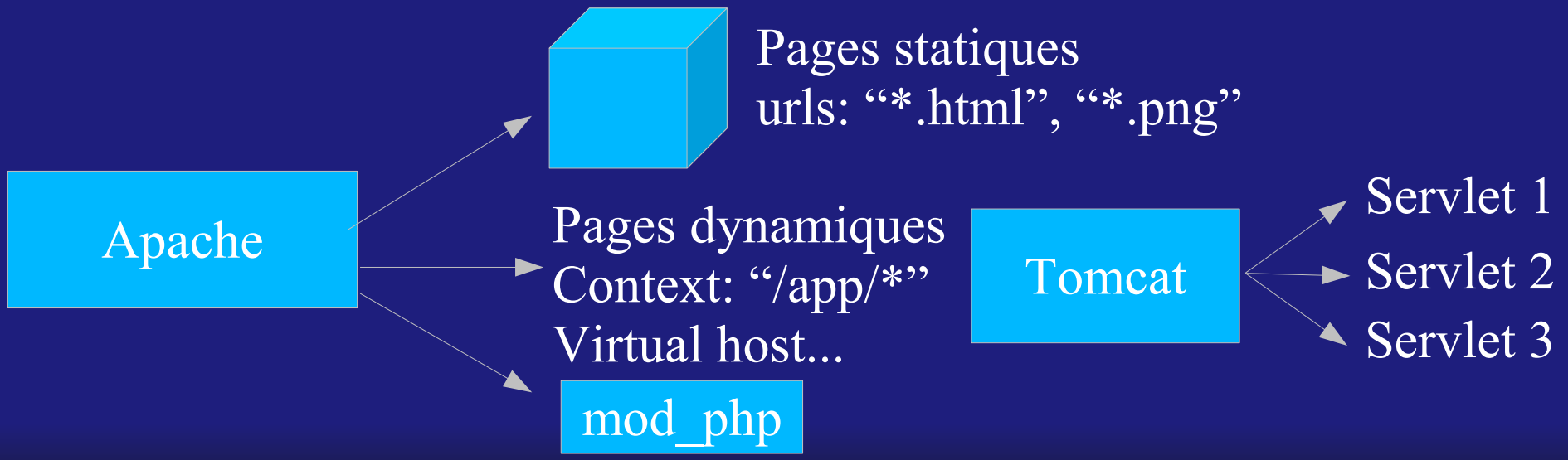
    … tout sous le répertoire “apache/web/”
  - index.html = liste des fichiers

- 60% des sites web utilisent Apache en frontal

SUNGARD

Apache:Static –> Tomcat:Dynamic

- Apache = server puissant pour pages statiques...
- redirige les contenus dynamiques vers des modules:
  - mod_php pour PHP, mod_python, ruby...
  - Tomcat pour Java jsp, Java Servlet …
  - autres…ex mod_svn



Utilisations de HTTP (2)

- HTTP n'impose presque rien...
- Input:
  - Nom resource + arguments &text=value
  - headers libres : text=value
- Output:
  - Headers libres : text=value
  - Text libre

Utilisations de HTTP (3) : Cookies

- HTTP est un protocol Déconnecté, Stateless
- ... Mais on le rend Stateful via les Cookies
- Le serveur
  - génère un id unique par client (en anglais “sessionId”)
  - Garde en mémoire/db le context du client
- Le client
  - garde le cookie du server
  - le retransmet à chaque appel
- Expiration de session~ logout/déconnection

SUNGARD

Utilisations de HTTP (4) Tunnels, TCP over Http, SOAP...

- HTTP en général non filtré par Firewalls
  - Pour des raisons historiques (doc, sites...)
  - Protocol en text clair, peut puissant
  - Protocol request-reply (client->server)
- HTTPS ~ Http ...mais crypté
- brèche réseau ...
  Utilisation de http comme couche transport
  = niveau 3 OSI (!= 7)
- Nombreux protocoles ré-implémentés sur Http (ex: Webservices= rpc/rmi, WS-sec, WS-XA,WS-event,...)

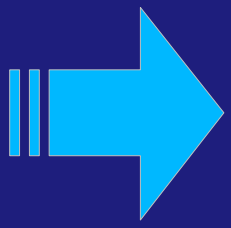
Limites et extensions de HTTP

- Http est un protocol simple, peu efficace!
  - une connection par appel!
    … possibilité de négocier par options, pour garder la socket ouverte
  - Non compressé
    … possibilité de négocier
  - Non sécurisé (authentification, cryptage, …)
    … possibilité de négocier, pour https
  - Stateless
    … possibilité de cookies, sessionId
  - Pas de QOS, broadcast, multicast...

Contenu Dynamique coté server

- Introduction, Html

- Protocole Http



- Contenu Dynamique côté server
  - Cgi-bin => Java Servlets, JSP
  - Apache, Tomcat, Architecture web
  - Frameworks MVC

- Contenu Dynamique côté client

Http et Contenus Dynamiques

- Historiquement, pseudo fichier “index.html”
  … = commande “ls -l”, si fichier non présent
- Premiers besoin de contenus dynamiques:
  - afficher l'heure, compteurs
  - changer de langues, customiser par clients...
  - fonctions de recherche
- Introduction des scripts “/cgi-bin/scripts.sh”
  - Fichiers déposés sur server, directement dispos
  - outils shell

Inconvénient portabilité cgi-bin

- Inconvénient majeur des cgi-bin : le shell
- Non typé, pas compilé, peu portable...
- combiné à des commandes ésothériques
  perl, sed, awk, cut, xargs, cat, grep, python, ...

- La maintenance des sites web est problèmatique

- Besoin d'un IDE, de gestion de projets, de tests, de processus de suivi de qualité …
  … d'un langage pour développeurs

Servlet … page en langage Java

- Coder une page en servlet = très simple :

```
import javax.servlet.http.*;
public class HelloWorldServlet extends HttpServlet {
   public void doGet(
         HttpServletRequest request,
         HttpServletResponse response) {
     response.setContentType("text/html");
     PrintWriter out = response.getWriter();
   out.println("<html><body>Hello</body></html>");
   }
}
```

Servlets

- Les Servlets … remplacent les cgi
  - Très légers en ressource
    directement appelés dans le serveur web
  - Appel fonction << fork de processus shell
- Server Web = “Container”
  - gère des pools de ressources
  - Alloue des Servlets/Threads à des Sockets
- De nombreux modules / langages existent
  … principes identiques (asp, php, ...)

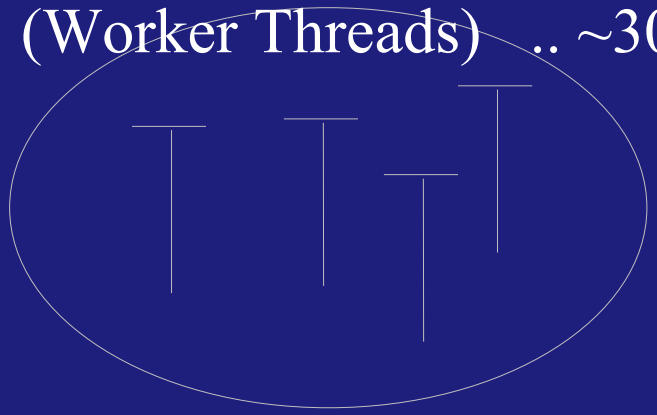
SUNGARD

Serveur Web = Container

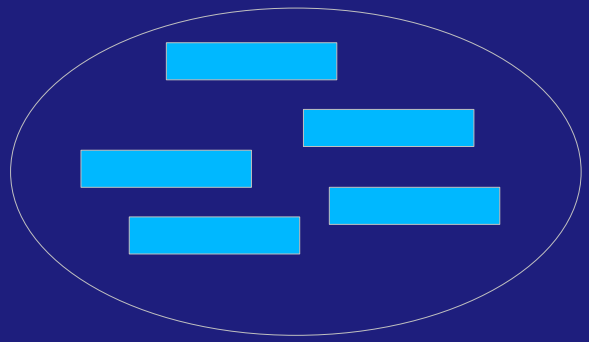
Listen thread

Listen socket

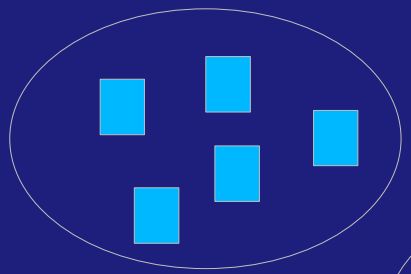
Pool de Threads (Worker Threads) .. ~300



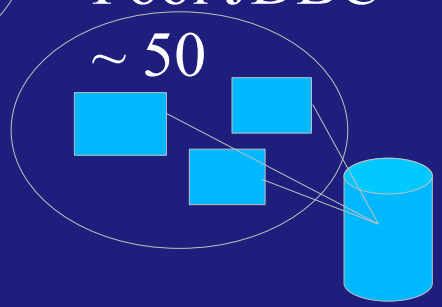
established sockets … max ~5000



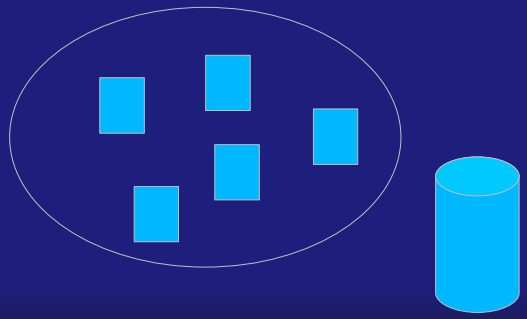
Pool de Servlets



Pool JDBC ~ 50

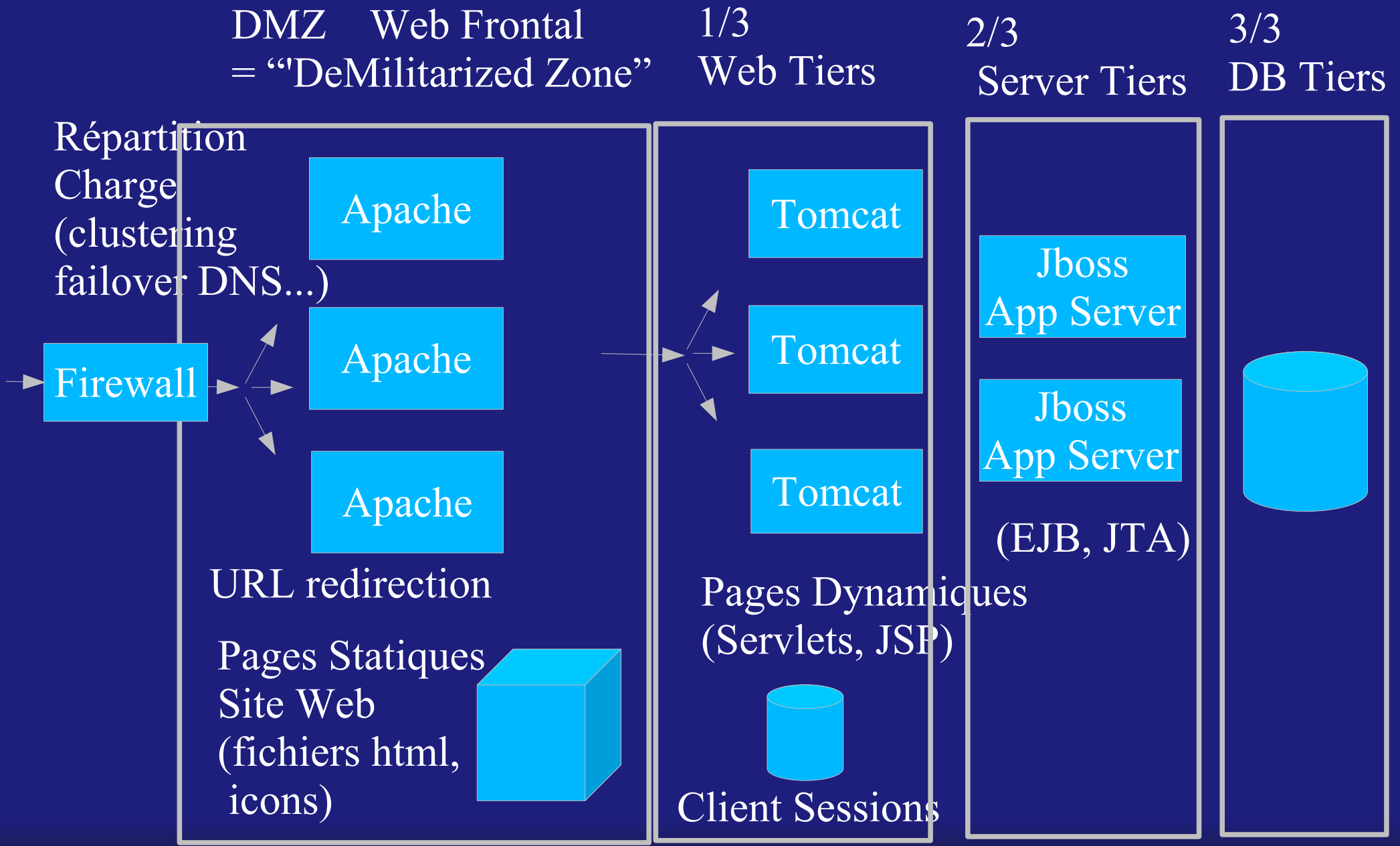


Alive Sessions... expiring (sessionId->Context) … 5000-100M



SUNGARD

Architecture Web Typique



Servlet … en détail

- plusieurs method pour http: doGet, doPost, … en général on utilise simplement doGet()
- Paramètres de doGet():
  - HttpServletRequest request
    … accès au flux socket input,
    aux paramètres url,
    et aux headers (=> aux cookies)
  - HttpServletResponse response
    … accès au flux socket output,
    et aux headers (=> aux cookies)

Servlet .... Text = Code

- En servlet, tout texte html est un bout de code

```
out.print(“<h1>un titre </h1>\n”);
out.print(“<A img=\\”icon.png\\”>img</A>”);
```

- C'est assez lourd...
  - Toute ligne, même vide s'écrit
    out.println(“”); ou out.print(“\n”)
  - Caractères spéciaux escapés...
    ex : \\” pour “ (java) .... & pour & (html)

Surcouche de Servlet : JSP

- JSP : outils “dual” des Servlets
  - pour simplifier les parties “textes”
  - Accès au java, via “escape” : <% %>
  - Factorisation du text via “includes” : tag-libs

- Pour les aficionados de PHP
  “nous aussi, on peut afficher une date dans du texte... et bien plus”
- Implémentation :
  page JSP = convertie+compilé en Servlet

Exemple de conversion JSP → Servlet

- En JSP (text.... contenant du java)
  - <h1>Hello “world”<h1> … text
  - <%= new Date() %> ... expr toString
  - <% for(int i=0;i<10; i++) {
%> Hello <%}%> ... stmt fragment
- => En servlet: (java... contenant du text)
  - out.print(“<h1>Hello \\”world\\”<h1>\n”);
  - out.print(new Date().toString());
  - for(int i =0; i<10;i++) {
out.println(“Hello”); }

Problèmes des JSP...

- JSP possède des inconvénients:
  - Syntax non xml <% %> <%= %> <@ ..>
  - Document ni valides en html, ni en xml !
  - Besoin d'éditeurs ad-hoc
- Par défaut, pages JSP non compilées... :(
- On PEUT et il FAUT les compiler :)
  - Sinon, surcoùt à la 1ere utilisation
  - Détection des pbs au runtime

Problèmes des JSP...

- La plupart des problèmes sont gérés par Eclipse, ou tout autre IDE web évolué
  - Editeur riche de la syntaxe html/jsp
  - Coloration syntaxique, autocompletion,
  - Erreurs syntaxiques à la compilation
  - Débuggueurs riches, intégrés avec java
  - ....
- Maven aussi sait bien gérer les JSP …
  - Of course!

Les Technologies Java MVC

- Basé sur les Servlets, JSP...
  très nombreux frameworks java

- simplifie l'utilisation des formulaires web (method http post + form + params)
- gestion des champs de saisis, paramètres, obligatoires, cookies
- sécurité, sessions, redirections, historique...
- … CMS évolués …

http://en.wikipedia.org/wiki/Mode-view-controller#Java

- Struts
- WebWork
- Tapestry
- Wicket
- Cocoon
- Spring MVC Framework
- Click Framework, Aranea, Induction, JSF, Oracle Application Framework, Sling, Stripes, WebObjects, , PureMVC, LongJump, Sofia, Struts2, Beehive...

100000 Frameworks Web

- Profusion de frameworks = aveu des faiblesses
  - Soit ils ne font pas bien les choses
  - Soit ils ne font pas assez de choses
  - Soit ils en font trop, trop compliqués
- C'est effectivement un peu des 3...

- Historiquement, Struts était novateur,... les autres n'ont pas révolutionné grand choses...

Pourquoi … Quel Issue ?

- frameworks QUE pour technologies servers (java, xml, xsl)
  … C'est la moitié du problème!

- Manque , Immaturité technologique sur client? Java, Javascript, Flex?

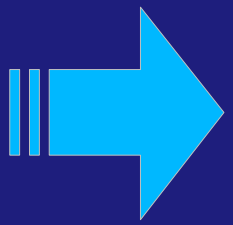
- Il manque du VRAI java coté client...
- Microsoft luttait contre Java coté client (affaires des procès contre Sun)… pas dans IE

Contenu Dynamique Coté Client

- Introduction, Html

- Protocol Http

- Contenu Dynamique coté server



- Contenu Dynamique coté client
  - JavaScript
  - Google GWT

Client-Side Web Technologies

- Historiquement...
  - Java était destiné aux petits appareils embarqués
  - Puis aux client web (pour l'aspect sécurité)
  - Puis finalement aux serveurs
- Applets Java = bonne idée pour intégrer Java au browser + http, mais il fallait le normaliser
- Netscape a normaliser JavaScript
- Pied de nez... Google remet Java via GWT, via Javascript, via Server !!! :)

JavaScript … Dynamic, Client-Side

- Html est un format static de rendu graphique
  - Refresh tout-ou-rien sur client
  - Refresh via Ajax … nécessite des appels réseaux client-serveur !
  - .... Refresh incrémental : Dhtml, DOM
- Netcape a inventé le “JavaScript”, et l'a fait normalisé...
  - Langage évalué coté client
  - Intégré à Netscape (devenu Firefox)
  - standard industriel de-facto

Principes du JavaScript

- Java ?
  - Reprend la syntaxe du C++/Java
    avec des “{“, des “for”, et des “;” …
  - Mais pas de classes, ni d'introspection!
- … Script !
  - Langage faiblement typé
  - Ne contient que des types simples
    (int, float, string, array …)
  - Non compilé, évalué au runtime

JavaScript : Script for client-side

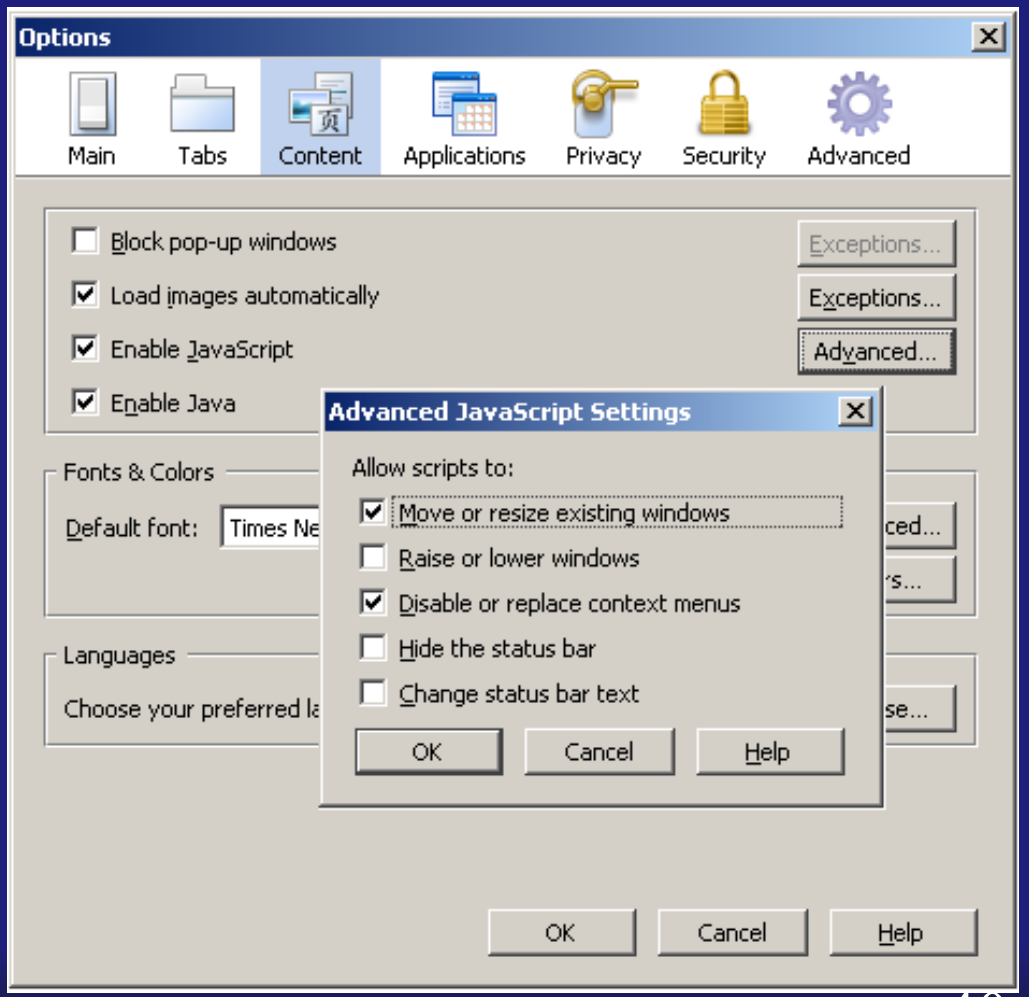
- évalué coté client = dans le browser
- Context d'évaluation rattaché à une page web
- transmis en text dans une balise html
  <script>function javascriptFunc() … </script>

- Accès au “Document Objet Model” de la page
  DOM = read/write page elts = Dhtml
- Callbacks graphiques (mouse,button ...)
- Libraire runtime du browser
  … Ajax threads, connections, popups, ...

SUNGARD

Enable JavaScript? Block Popups

- Historiquement, il était courant de “bloquer les scripts javascripts” pour tous les sites...
- Annoying popups!
- ... security holes !
- Still security pb!
- Rule
  “No cross site scripting”



JavaScript: abération de l'histoire ou futur du web ?

- Aujourd'hui, les JavaScripts son INCONTOURNABLES
- moyen de faire des clients riches...
  avec des technologies web!

  (alternatives, ex: Adobe Flash + Flex)
- Largement répandus... et promus par les acteurs majeurs du web:
  - Google, Firefox(Mozilla), Microsoft...

GWT

- Google change TOUT avec son
  Google Web Toolkit
- Java coté Server ET coté Client ...
- Techniquement:
  - Java est “compilé en javascript” pour les parties clientes !!!
  - riche librairie d'abstraction graphique
  - On fait du web comme on fait du swing

Analogies Swing - GWT

- La plupart des composants UI existent en GWT
- Ex de code swing:

```
root = new Jpanel();
root.setLayout(new BorderLayout());

JTextArea text = new JTextArea();
root.add(but, BorderLayout.CENTER);

JButton but = new JButton("click");
root.add(but, BorderLayout.SOUTH);
```

GWT listener callbacks (Client Side)

- Ex code Swing: inner classes callbacks

```
jbut.addActionListener(new ActionListener() {
  public void action(ActionEvent evt) {
    // my callback for click button
  }
});
```

- Equivalent GWT .... idem !

GWT : RPC - Servlet

- GWT simplifie aussi les appels Client->Server
- Impression d'appeler une method java remote
  ... implémenté en JSON+Http-RpcServlet+...

- Mise en oeuvre très simple: 3 classes
  - Interfaces + Implémentation + Interface Async

- code multi-threading robuste, sans piège
  … car les appels sont forcément en “Async”
  … en respectant le Thread UI + Pool 2 Threads

GWT Debugging

- débuggueur très pratique
- en mode “Hosté” (développement):
  - pas de JavaScript!
  - que du java, même pour le client
    (implémentation en double du runtime GWT via Chrome + méthodes natives)
- en mode réel
  - javascript généré pour le client
  - Pas de débuggueur mais un viewer
  - Sniffer protocol http

GWT !

- GWT comble un manque sur les technologies web, coté client et server
- Javascript est caché, mais reste accéssible
- De même css, html, http...
- A terme, peut-être que javascript sera remplacé par des applets, dans Chrome?
- … à suivre

Questions

Questions ??

arnaud.nauwynck@gmail.com