

# Αρχιτεκτονική προηγμένων web εφαρμογών

---



Σαράντης Πασκαλής <paskalis@di.uoa.gr>  
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

# Προβλήματα CGI, SSI

- CGI: Απόδοση
  - Έναρξη νέας διαδικασίας για εκτέλεση (2 διαδικασιών σε περίπτωση διερμηνευόμενης γλώσσας, π.χ. perl).
  - Κοινές διαδικασίες αρχικοποίησης πρέπει να εκτελούνται κάθε φορά.
- SSI:
  - Παρόμοια προβλήματα όταν χρησιμοποιεί CGI. Επιπλέον επεξεργασία για την ανάλυση SSI σελίδων.
  - Κίνδυνος ασφάλειας χωρίς λεπτομερείς ρυθμίσεις.
  - Μη κλιμακώσιμο και μη επαναχρησιμοποιήσιμο.



# Εγγενή APIs

- NSAPI, ISAPI (Netscape, Microsoft)
  - Μηχανισμοί που επιτρέπουν την απευθείας πρόσβαση στο API του server.
  - Χρήση βελτιστοποιημένου κώδικα για τις συγκεκριμένες εργασίες.
  - Δεν υπάρχουν προτυποποίηση, είναι διαφορετικές προσεγγίσεις, άρα δεν είναι δυνατό να δημιουργηθούν φορητές εφαρμογές.



# FastCGI

- Προσπάθεια συνδυασμού φορητότητας CGI με αποδοτικότητα εγγενών APIs.
- Επιτρέπει στις διεργασίες CGI να παραμένουν «ζωντανές» και μετά την εξυπηρέτηση μιας αίτησης.
- Απαιτείται η δημιουργία τμήματος (module) στο server για την λειτουργικότητα FastCGI και την επικοινωνία με το server.
- Η προτυποποίηση FastCGI δεν πρόλαβε να εξελιχθεί και να δημιουργηθούν τα κατάλληλα modules για τους δημοφιλείς servers.



# Επεξεργασία μορφότυπων (Template Processing)

- Προσέγγιση που στοχεύει στην απλούστευση της δημιουργίας δυναμικού περιεχομένου.
- Templates: HTML σελίδες με επιπρόσθετες ετικέτες (tags), που προδιαγράφουν μεθόδους για δυναμική εισαγωγή παραγόμενου περιεχομένου από εξωτερικές πηγές.
- Δημοφιλή προϊόντα: PHP, Allaire/Macromedia Cold Fusion, Microsoft ASP.



# Παραδείγματα PHP

```
<?php  
echo 'Hello, World!';  
?>
```

```
<?php  
//statements here  
?>  
regular html here  
<?  
//more php statements  
?>
```



# Παράδειγμα ASP

```
<html>
<body>

<%
    response.write("Hello World!")
%>

</body>
</html>
```



# Παράδειγμα Cold Fusion

```
<CFQUERY NAME="query1" DATASOURCE="oracle" ...>
  SELECT id, columnX, columnY, columnZ
  FROM TABLE1
  WHERE id = #substitution-parameter#
</CFQUERY>
<CFIF query1.recordcount GT 0>
  <TABLE>
    <CFOUTPUT QUERY="query1">
      <TR>
        <TD>#columnX#</TD>
        <TD>#columnY#</TD>
        <TD>#columnZ#</TD>
      </TR>
    </CFOUTPUT>
  </TABLE>
</CFIF>
```





# Επεξεργασία μορφότυπων (Template Processing)

- Βελτιωμένη τεχνική SSI.
- SSI → απλές εργασίες.
- template processing → σύνθετες εργασίες
  - υποβολή ερωτήσεων σε βάσεις δεδομένων
  - επαναληπτική επεξεργασία (while, for loops)
  - υποθετική επεξεργασία (δηλώσεις if)



# Επεξεργασία μορφότυπων (Template Processing)

- Πλεονέκτημα: Είναι διαχειρίσιμες οι σελίδες από σχεδιαστές με υπόβαθρο σε HTML και γραφικά, αλλά όχι προγραμματισμό. Η χρήση επιπλέον ετικετών είναι απλούστερη από γνώση προγραμματισμού.
- Πρόβλημα: Όσο πιο προηγμένες είναι αυτές οι τεχνικές, τόσο περισσότερο προσομοιάζουν σε γλώσσες προγραμματισμού → η επιθυμητή απλότητα δεν επιτυγχάνεται.



# Διαχωρισμός Περιεχομένου και Παρουσίασης

- Περιεχόμενο: επεξεργασία αναγκαία για την ανάκτηση περιεχομένου
- Παρουσίαση: παράμετροι σχεδιασμού για την παρουσίαση
- Η μίξη τους σε ένα τμήμα κώδικα κάνει δυσκολότερη τη συντήρηση του (προγραμματιστής; σχεδιαστής;)
- Ιδανικά: προγραμματιστές να διαχειρίζονται την επεξεργασία για περιεχόμενο, και οι σχεδιαστές την παρουσίαση.
- Model-View-Controller σχεδιασμός.



# Servlets

- API βασισμένο στην τεχνολογία Java.
- Φορητές εφαρμογές μεταξύ διαφορετικών λειτουργικών συστημάτων και πλατφόρμας.
- Χρησιμοποιεί επαναχρησιμοποιήσιμα και μόνιμα server modules (όπως το FastCGI).
- Το API είναι φορητό μεταξύ servers, λειτουργικών συστημάτων, πλατφόρμας υλικού (αντίθετα με το FastCGI).
- Τα servlets εκτελούνται πανομοιότυπα σε οποιοδήποτε περιβάλλον παρέχει συμβατό servlet runner.



# Servlets

- Δεν απαιτείται γνώση APIs υποκείμενων servers. Η διεπαφή με τους servers είναι ευθύνη των servlet runners.
- servlet runner: περιλαμβάνει JVM, επικοινωνεί με HTTP servers
  - απευθείας μέσω JNI
  - τρέχοντας αυτόνομα και «ακούγοντας» για προωθημένες αιτήσεις από servers.



# Java Server Pages (JSP)

- Αρχικά: template processing με servlets.
- Σήμερα: οι επεξεργαστές JSP παίρνουν JSP σελίδες, τις μεταφράζουν σε servlets και στη συνέχεια σε Java bytecode. Ουσιαστικά το JSP είναι προ-επεξεργαστής για servlets.
- Αποτέλεσμα: μεταγλωττισμένα modules που εκτελούνται γρηγορότερα από διερμηνευόμενα.



# Java Server Pages (JSP)

- Οι εντολές JSP είναι ετικέτες XML, συνδυασμένες με τμήματα κώδικα Java.
  - Ο συνδυασμός κώδικα και ετικετών εκφράζει την υπηρεσία που παρέχει το τελικό αποτέλεσμα.
- Δεν απαιτείται όλη η λειτουργικότητα να βρίσκεται σε JSP.
- Παρόμοια λογική με SSI: διαφοροποιείται η παρουσίαση από την υπηρεσία, υποστηρίζεται επαναχρησιμοποιήσιμος σχεδιασμός.



# Java Server Pages (JSP)

- Γενικότερης χρήσης από servlets.
- Αποδοτικότερη και μη επικίνδυνη λύση σε σχέση με SSI.





# Μελλοντικές Κατευθύνσεις

- Επιπλέον αποδέσμευση παρουσίασης και υπηρεσίας.
- Model-View-Controller σχεδιασμός
  - model → το λογικό μοντέλο της εφαρμογής
  - view → η παρουσίαση
  - controller → η οντότητα που ελέγχει τις αλληλεπιδράσεις μεταξύ των διαφορετικών τμημάτων.
- Ευελιξία: π.χ Αλλάζει η παρουσίαση, χωρίς αλλαγή των άλλων τμημάτων.
- Struts Application Framework



# Μελέτες σύγκρισης (1)

- Titchkosky, L., Arlitt, M., and Williamson, C. A performance comparison of dynamic Web technologies. *SIGMETRICS Perform. Eval.Rev.* 31, 3 (Dec. 2003), 2-11.
- Η δυναμική παροχή υπηρεσιών υποβιβάζει κατά 8 φορές την απόδοση από
  - πρόσβαση σε βάσεις δεδομένων
  - επεξεργασία δυναμικού περιεχομένου
- Κάθε τεχνολογία έχει τα αδύνατα σημεία της
  - PHP: καλή για μικρές δυναμικές αιτήσεις, προβληματική για μεγαλύτερες
  - Java based (jetty, resin, tomcat): ακατάλληλη για στατικό περιεχόμενο, καλή για δυναμικό
  - Γενικά: οι τεχνολογίες Java αποδίδουν καλύτερα από Perl/PHP.
- Η απόδοση για μερικές τεχνολογίες είναι απρόβλεπτη σε μεγάλο φόρτο.



# Μελέτες σύγκρισης (2)

- V. Apte, T. Hansen, and P. Reeser. Performance Comparison of Dynamic Web Platforms. Computer Communications 26(8), May 2003.
- Σε πολύπλοκες επεξεργαστικά εφαρμογές εμφανίζονται προβλήματα της τεχνολογίας Java.
- Σε απλούστερες επεξεργαστικά εφαρμογές, οι τεχνολογίες Java κλιμακώνονται πολύ αποτελεσματικότερα.
- Η αποδοτικότερη τεχνολογία ήταν ο συνδυασμός FastCGI/C++.
- Η χρήση servlets αντί για JSP υποβάθμιζε την απόδοση.



# Μελέτες σύγκρισης (3)

- G. Gousios, and D. Spinellis. A comparison of portable dynamic web content technologies for the apache web server. In Proceedings of the 3rd International System Administration and Networking Conference SANE 2002, pages 103–119, May 2002.
- FastCGI: η αποδοτικότερη τεχνολογία
- PHP: Δεν κλιμακώνεται σωστά (οι χρόνοι απόκρισης έχουν μεγάλη διασπορά)
- servlets: Πιο αργή γενικά, αλλά με αναμενόμενους χρόνους απόκρισης.

