

# Struttura del seminario

- Premessa. Chi siamo noi. Chi è Alice.
- Cosa ottimizzare? La regola del 10/90. Tecniche di misurazione dei tempi.
- Tecniche classiche di ottimizzazione java
- Cache di contenuti. Quali scegliere e quali tecniche adottare.
- Cache dell'html. Perché decidere di cachare un'intera pagina html dinamica
- Comprimere le pagine html. Come? Quando? Perché?
- Multithreading. Come sfruttarlo per migliorare le performance. Pro e contro.
- Spazio alle domande: question time

# Presentazione di K-TECH

**K-TECH** opera in Italia nel settore della consulenza e dei servizi tecnologici offrendo assistenze tecniche specialistiche nel campo delle applicazioni Java Enterprise, Wireless e sulla piattaforma Oracle.

## Siamo partner di:



## Aree di intervento

- Educational
- Consulting
- Projects

## Siti ufficiali:

- [www.oracleportal.it](http://www.oracleportal.it)
- [www.javaportal.it](http://www.javaportal.it)



## Alcuni clienti:

- Q8 Kuwait Petroleum Italia SpA
- Gruppo Vitrociset SpA
- Epiclink SpA
- Bea Systems Italia S.p.A.
- World Food Programme
- Alcatel Spaces Industries
- KPMG Consulting S.p.A.
- Gruppo Engineering S.p.A.
- Sun Microsystem Italia
- Borland S.p.A.
- Telecom Learning Services
- Finsiel S.p.A.
- Datamat S.p.A.
- IBM - Selfin S.p.A.

## Collaboriamo con:



# Presentazione di RossoAlice

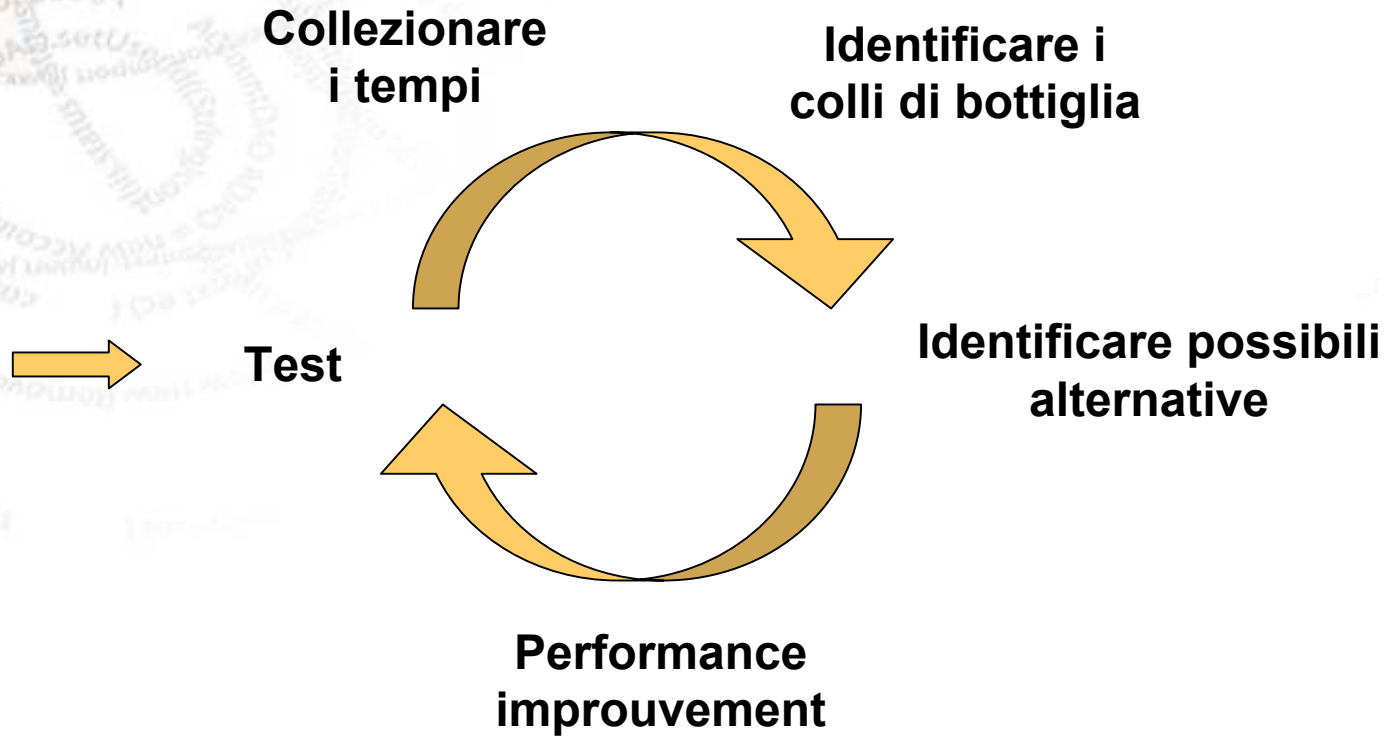
Il portale Rosso Alice è un portale multi canale dedicato ad un vasto numero di visitatori. Con Alice è possibile acquistare suonerie, canzoni, video, giochi, film, leggere le ultime notizie, giocare online, vedere le anteprime di film e notizie sportive e molto altro ancora.

## Alcuni numeri di RossoAlice

- 7 Web server Apache
- 24 nodi BeaWeblogic
- 5 server Broadvision
- 4 database Oracle
- 1 dominio distribuito su 3 macchine Sun
- 9 ambienti di sviluppo e 3 di test
- 3 team di lavoro: Roma, Milano e Vicenza



# Metodologia di performance tuning



# Cosa ottimizzare?

## La regola del 10 / 90

La regola empirica del 10 / 90 afferma che:

*“Il 10% per codice non ottimizzato viene eseguito per il 90% del tempo.”*

### Vantaggi

- Attenzione del programmatore mirata ad una quantità di codice ridotta
- Piccole ottimizzazioni possono migliorare significativamente le performance del sistema

# Tecniche di misurazione dei tempi

## Misurazione da codice java

Pro

- Economico
- Facile da implementare

Contro

- Difficoltà di collezionare i risultati ottenuti
- Difficile monitorare i tempi di classi già compilate

## Misurazione con tools di performance tuning

### *ServerTrace*

Pro

- Grande facilità di interpretazione dei dati
- Semplicità a monitorare i tempi di classi già compilate

Contro

- Costi elevati
- Necessità di conoscere un nuovo tool

## Tool di stress

### *Jmeter, WebStress, Grinder...*

Pro

- Se ne trovano di gratuiti
- Permettono di stressare la macchina da un solo client

Contro

- Danno un'indicazione dei tempi di risposta ad una richiesta senza entrare nel dettaglio
- I risultati sono affetti dal tempo di trasferimento dati sulla rete

# Tecniche classiche di ottimizzazione java

## Stringhe

- A runtime utilizzare append di StringBuffer invece dell'operatore +
- A compile time scegliere +

## I/O

- Limitare l'uso di classi che fanno i/o un byte alla volta
- Utilizzare buffered i/o

## Cicli

- Calcolare gli invarianti prima del ciclo
- Limitare le chiamate ai metodi al minimo indispensabile

## JDBC

- Pool di connessioni
- PreparedStatement
- SELECT mirate
- Utilizzare gli indici delle tabelle

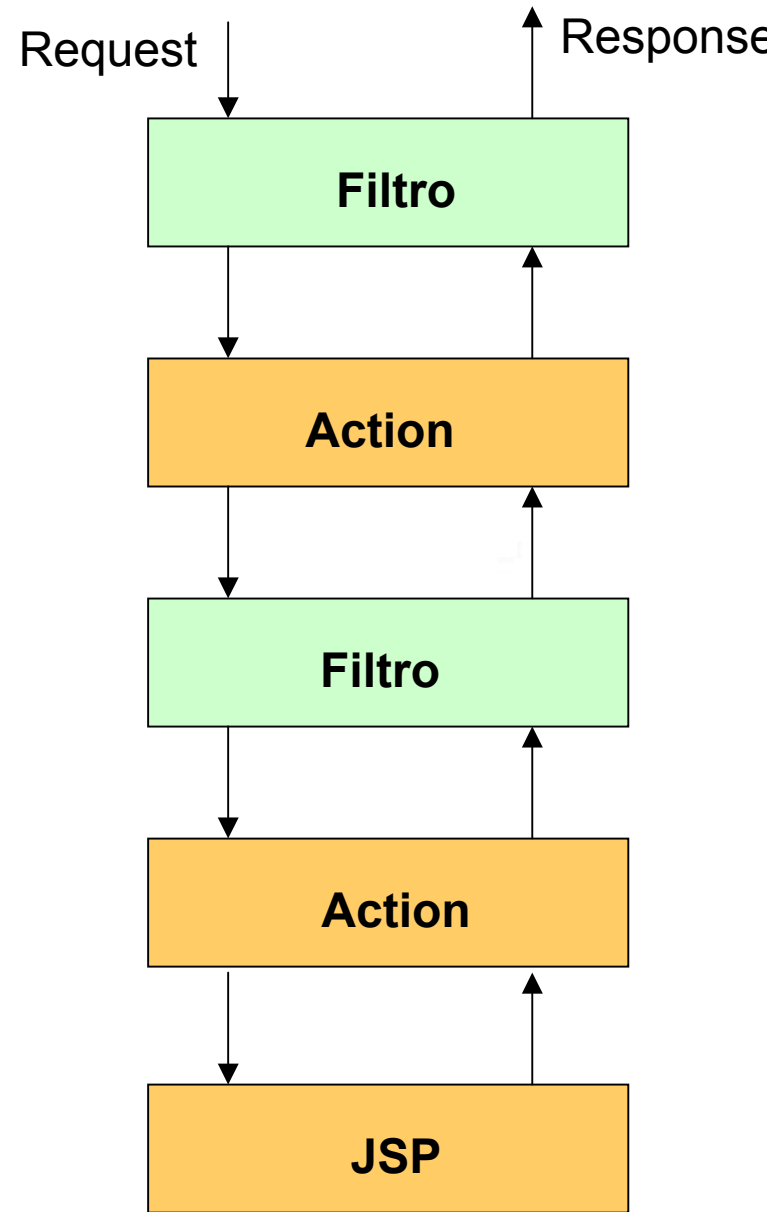
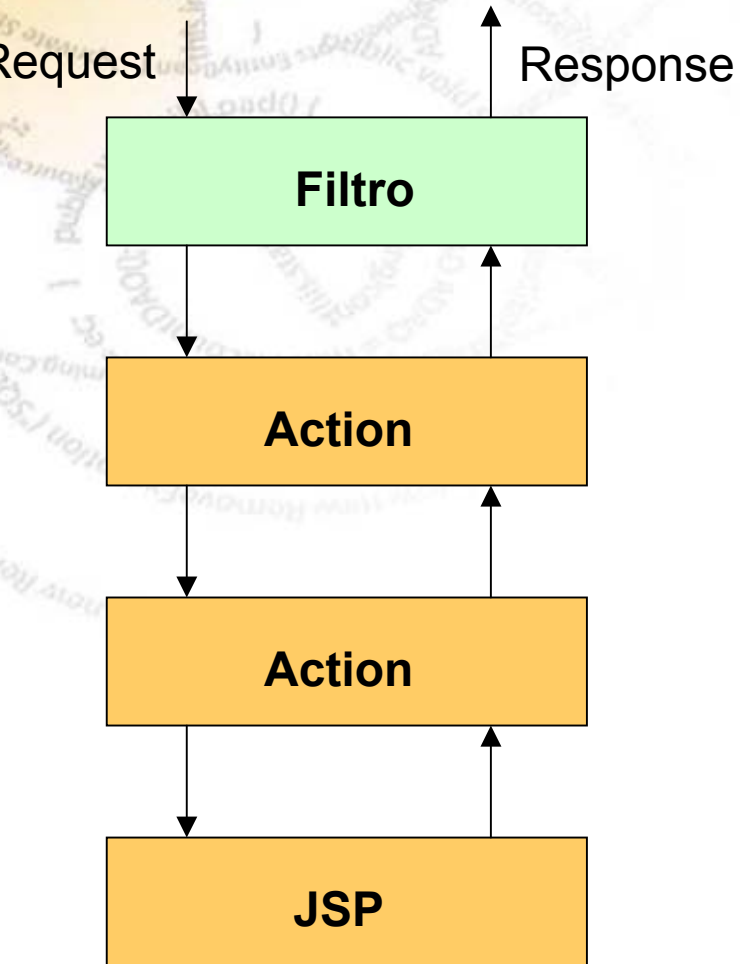
## Collections

- Scegliere il tipo più adatto in funzione del tipo di operazione prevista
- Limitare l'uso di oggetti sincronizzati solo quando necessario

# Collections

Type of operation	Adding objects at end	Adding objects at middle	Adding objects at first
<b>ArrayList</b>	Fast	slow (slower than when adding objects at last)	slow (slower than when adding objects at last and middle)
<b>Vector</b>	fast (slower than ArrayList because of synchronization)	slow (slower than when adding objects at last)	slow (slower than when adding objects at last and middle)
<b>LinkedList</b>	fast (but slightly slower than ArrayList and Vector)	worse(worse than every operation)	slow (slower than when adding objects at last and middle)

# Filtri



# Compressione HTML

## Perché usarla?

Perché riducendo la quantità di dati da spedire diminuisce il tempo di trasferimento. Un codice html formattato con spazi può essere compresso anche con un fattore 10:1.

## Come usarla?

Sviluppando un filtro java (`javax.servlet.Filter`) che intercetta il codice html e prima di inviarlo al client lo comprime.

Come alternativa si può usare un modulo di compressione del webserver.

## Svantaggi?

Se non usata con un meccanismo di cache dell'html può richiedere una quantità significativa di risorse macchina a causa dei complessi algoritmi di compressione.

# Cache

*“Caches are effective because it is expensive to move data from one place to another or to calculate results.” (Jack Shirazi)*

## Campi di applicazione

- Query a database
- JNDI lookup
- Lettura di file
- Accessi a risorse non locali
- Risorse il cui ricalcolo è molto dispendioso

## Progetti di cache

- Open Symphony cache
- JCS (Java Caching System) del progetto Turbine di Jakarta
- JCache di SouceForge

# Cache di contenuti

Per contenuti si intende qualsiasi oggetto candidato ad essere inserito nella cache (vettori con le voci di un menù, stringhe che descrivono il contenuto di parte della pagina html, record di tabelle statiche...)

## **Tecniche:**

- cache a tempo indeterminato o determinato
- aggiornamento in foreground o in background
- cache centralizzata o distribuita
- svuotamenti programmati della cache

# Cache della pagina html

Questa tecnica permette di minimizzare la costruzione di pagine dal contenuto dinamico.

E' la scelta adatta per la costruzione di pagine dinamiche il cui cambiamento avviene nel tempo.

Non è adatta per le pagine che dipendono dalla singola richiesta (ad esempio pagine di un motore di ricerca).

Può essere implementata con un filtro java che intercetta la response e la inserisce in cache. Richieste successive prendono l'html salvato nella cache invece di completare il giro standard per la costruzione dell'html.

Piccole modifiche possono essere apportate all'html prima di inviarlo al client (ad esempio per modificare il nome utente visualizzato nella pagina in funzione dell'effettivo utente loggato).

Nel progetto Rosso Alice si è fatto uso di questa tipologia di cache per la maggior parte delle pagine html principali del sito.

# Multithreading

*“Threaded programs create two or more simultaneous flows of control to accomplish multiple tasks at the same time” (Sun website)*

## Pro

- Permette di fare più operazioni contemporaneamente
- Massimizza il carico della cpu

## Contro

- Difficile da programmare correttamente
- Difficile individuare alcuni tipi di bug
- Gestione sincronizzata degli accessi alle risorse condivise
- Non adatto ai programmatori junior

## Quando usarlo?

- Operazioni con basso carico di cpu e lunghi tempi di attesa che sono indipendenti le une dalle altre (ad esempio lettura di file, query su database, invio e ricezione dati su socket...)
- Operazioni complesse (anche con alto carico di cpu) che possano essere suddivise in operazioni elementari indipendenti nel caso in cui si abbia a disposizione una macchina multiprocessore con poco carico

# Bibliografia

Il miglior libro sulle performance java:

- Java Performance Tuning 2nd edition di Jack Shirazi – O'Reilly

Siti sulle performance java:

- <http://www.javaperformancetuning.com>
- <http://java.sun.com/docs/performance>
- <http://www.precisejava.com/>

Articolo sui filtri java per la compressione e la cache dell'html:

- <http://www.onjava.com/pub/a/onjava/2003/11/19/filters.html>

Siti produttori tool di stress:

- <http://jakarta.apache.org/jmeter/>
- <http://grinder.sourceforge.net/>
- <http://www.microsoft.com/technet/itsolutions/intranet/downloads/webstres.msp>

# Ringraziamenti

- K-tech per la disponibilità a partecipare a questo evento
- Telecom Italia s.p.a. per la possibilità di illustrare alcune delle tecniche usate nel progetto “Rosso Alice”
- Massimo Cimino e Pino Galli che hanno fatto parte del gruppo performance e che hanno partecipato allo sviluppo di alcune delle tecniche illustrate nel seminario
- Gabriele Folchi che ha inizialmente coordinato il gruppo performance del vecchio portale di Telecom Italia “la casa di Alice” e che ha messo le basi per la nascita del gruppo performance di “Rosso Alice”
- Mario Pantaloni e Carla Timpani che ci hanno convinto a presentare questo seminario

# Riferimenti

Presentazione a cura di Mirko Scognamiglio (m.scognamiglio@k-tech.it)

Relatore Davide Lorenzo Marino (d.marino@k-tech.it oppure dl.marino@tiscali.it)