UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II
Facoltà di Ingegneria
Corso di Studi in Ingegneria Informatica

tesi di laurea

Analisi di strumenti e tecniche per lo sviluppo di applicazioni Ajax.

Anno Accademico 2006/2007

relatore
Ch.mo prof. Porfirio Tramontana

candidato
Antonio Pandolfo
Matr. 41/2568

Sommario

- Introduzione
- Problematiche
- Obiettivi
- Soluzione proposta
  - Tecnologie utilizzate
  - Esempi di utilizzo
- Conclusioni

Introduzione (1)

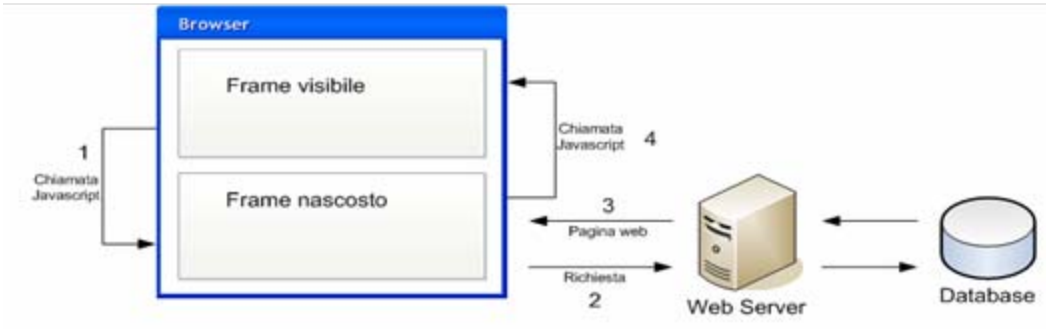
Il web 2.0 e Ajax

- Il web 2.0 sta rapidamente ridisegnando il web e il modo di interagire con esso. Alcune caratteristiche:
  - Applicazioni web simili a “Desktop Application”(una ricca ed interattiva, interfaccia user-friendly) .
  - La rete come piattaforma (rilascio di applicazioni tramite browser)
- Una delle tecnologie che rende possibile il web 2.0 è la comunicazione asincrona con il server che vede la sua più moderna e potente espressione nell’engine Ajax.

Introduzione (2)

Tecniche di comunicazione Asincrona.

- **Tecnica del frame nascosto**
  - Permette di mantenere la cronologia.
  - Non permette di saper cosa accade dietro le quinte. Se la pagina non si carica, all'utente non viene notificato alcun problema.
  - Si rende necessaria una tecnica che aggiri questo inconvieniente e formalizzi la chiamata asincrona.



Introduzione (3)

- **Oggetto XMLHttpRequest (il vero AJAX)**

  Nella comunicazione Client-Server viene inserito un livello intermedio rappresentato dall'engine Ajax. Quest'ultimo è nient'altro che un oggetto Javascript che viene chiamato in ogni comunicazione client-server.

  - **Vantaggi:**
    - Il codice è più pulito e gli intenti della varie operazioni sono più manifesti
    - Accesso agli header di richiesta e risposta.
    - Acesso ai codice http di stato cosi da poter determinare se la richiesta è andata o meno a buon fine.
  - **Svantaggi:**
    - Non vi è alcuna registrazione della cronologia.
    - In Internet Explorer bisogna attivare il controllo ActiveX per rendere disponibile l'oggetto XMLHttpRequest.

Problematiche

Problematiche relative alla realizzazione di applicazioni Ajax:

- L'utilizzo di Ajax richiede una grande quantità di codice aggiuntivo.
- Richiede la conoscenza di diversi linguaggi e tecnologie
  - Javascript, CSS, DOM ecc…
- Richiede che lo sviluppatore faccia i conti con le incompatibità tra i browser.
- Difficoltà nel debug.
- Il programmatore deve gestire i tasti back e forward.
- L'esposizione del codice Javascript sul client è potenzialmente rischiosa.

Obiettivi

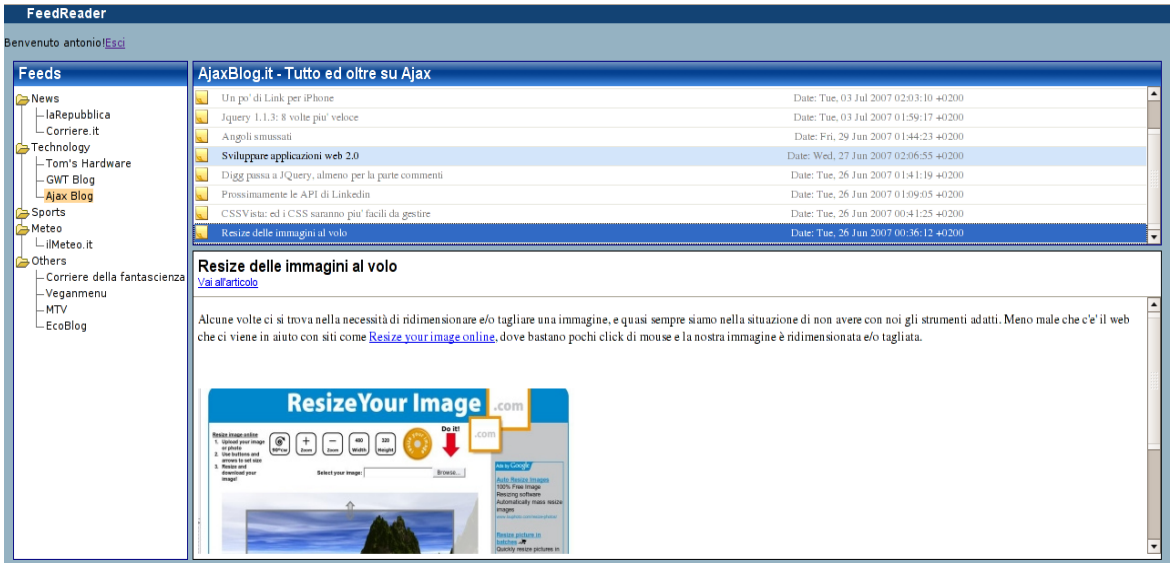
Studio comparato di alcuni frameworks (open source) oggi disponibili, mediante l'applicazione degli stessi ad una web application.

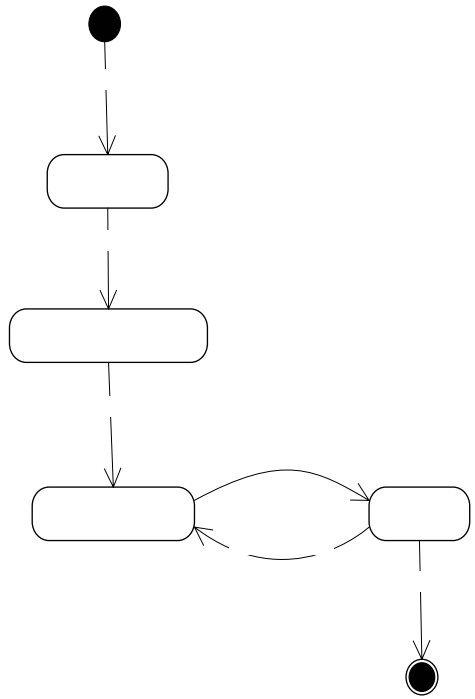
- Framework lato client: sono basati sul browser, si tratta per lo più di librerie Javascript.
- Framework lato server: sono basati sul server, intervengono nel modo in cui il server interagisce con il client (meccanismo della chiamata remota). Intervengono a vari livelli nella programmazione e progettazione lato server.

Obiettivi: FeedReader

E' una applicazione web per la lettura dei feeds Rss e Atom. Caratteristiche:

- Utilizzo di Javascript a oggetti
- Funzionamento e interazione con l'utente simili ad una classica applicazione desktop
- Layout e disposizione dei pannelli simile ai classici programmi di posta.





Soluzione proposta 1 : DOJO

Dojo è un framework Open Source che permette la creazione di funzionalità dinamiche nelle pagine web:

- **Fornisce un potente meccanismo di astrazione IO Ajax-based**. Questo significa che esso gestisce la comunicazione asincrona attraverso l'uso di XMLHttpRequest. Questo meccanismo è chiamato remoting.
- **Fornisce un potente gestore di eventi**. Per esempio è possibile invocare il gestore di eventi prima o dopo che un evento si verifichi.
- **Fornisce un costruttore dell'interfaccia utente basato su Widgets**. E' possibile costruire nuovi widgets o usare quelli costruiti da altri.
- **Supporto delle animazioni**. Sono disponibili un gran numero di utili librerie Javascript.

Soluzione proposta 1 : DOJO

Funzionalità implementate nell'applicazione FeedReader:

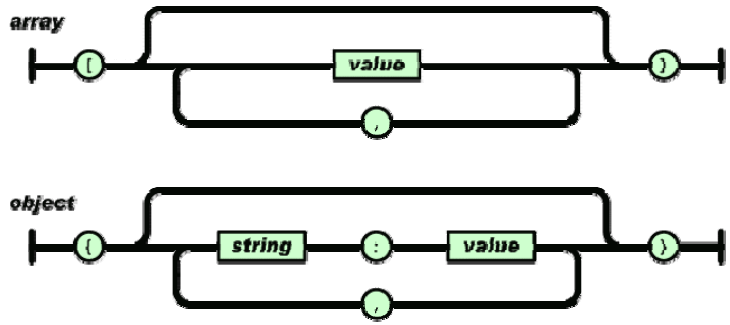
- Gestione della comunicazione asincrona e delle operazioni di basso livello sull'oggetto XMLHttpRequest attraverso il metodo dojo.io.bind()
- Utilizzo delle API specifiche per la manipolazione del DOM
- Possibilità di utilizzare direttamente il formato XML restituito dal server impostando il parametro mimetype
- Gestione del tasto Back
- Gestione degli eventi
  - Evento logout: pulitura dello schiermo e ripristino delle condizioni iniziali
  - Concatenamento di chiamate di funzioni.
- Utilizzo del widget Tree.

Soluzione proposta 1 : DOJO

Dojo e Json:

JSON (JavaScript Object Notation) è un semplice formato per lo scambio di dati. Per gli utenti è facile da leggere e scrivere, mentre per le macchine risulta facile generare e analizzare la sintassi

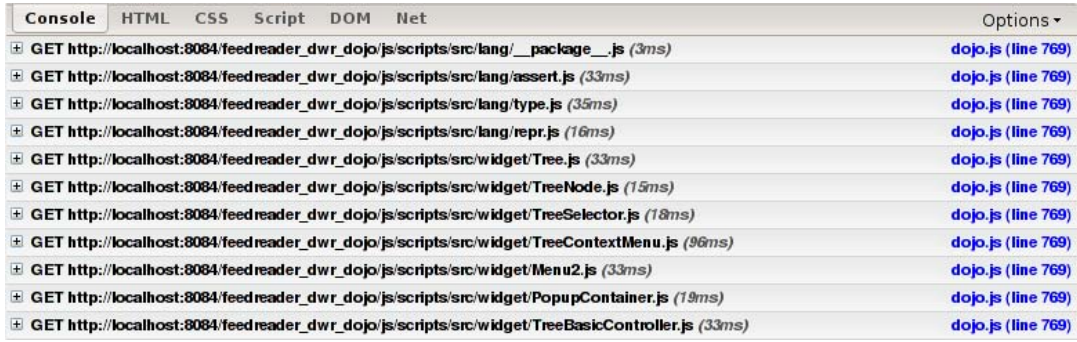
1feeds.xml	1,5 kB	XML document
1feedsjson.txt	1,3 kB	Documento in testo semplice

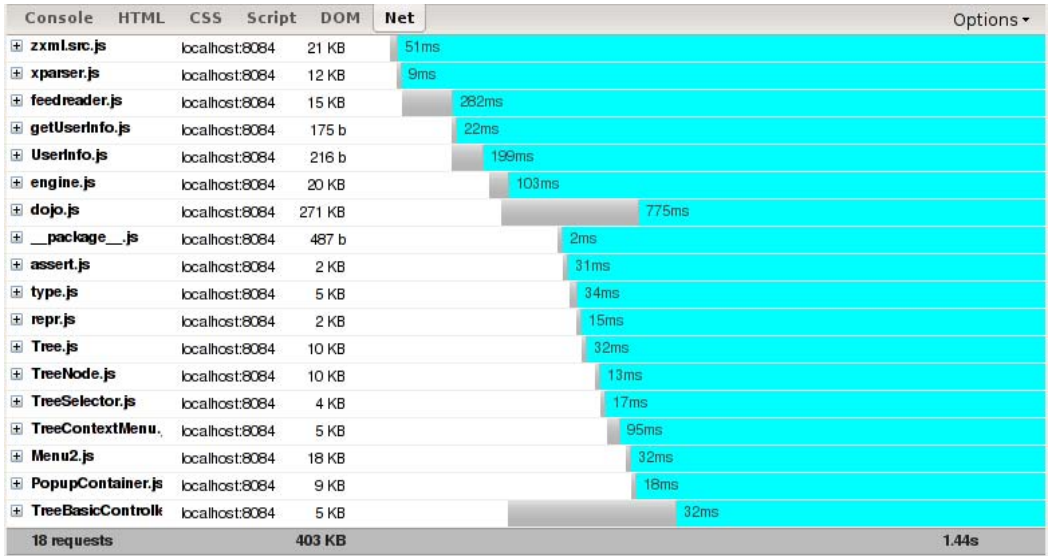


- Vantaggi:
  - Permette di ottenere un vantaggio nelle dimensioni dei dati scambiati con il server.
  - Rende piu rapido l'accesso ai dati restituiti dal server.
  - Rende più semplice la manipolazione dei dati restituiti dal server grazie al formato interpretabile da Javascript.
- Svantaggi:
  - Meno leggibile rispetto ad XML da parte di un utilizzatore esterno.

Soluzione proposta 1: Dojo

I tempi di caricamento sono accettabili a patto che non si esageri con l'utilizzo di effetti grafici.

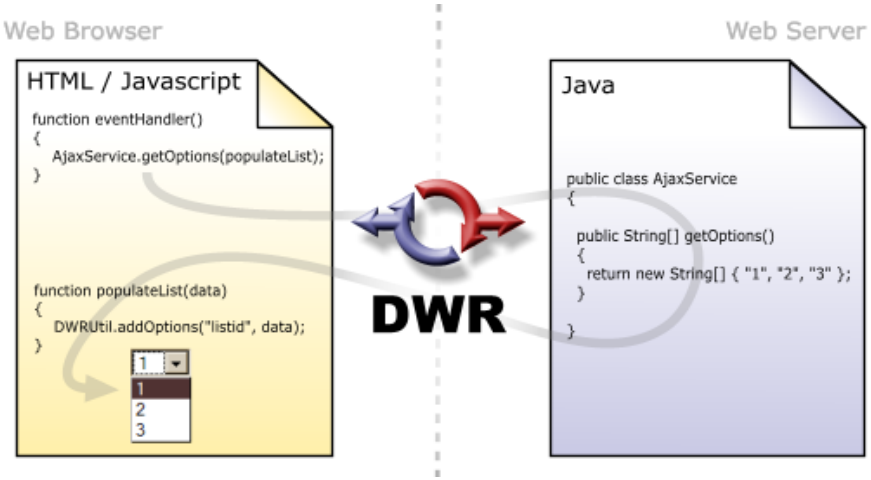




Soluzione proposta 2 : DWR

Frameworks “RMI-like Remoting via Proxy”

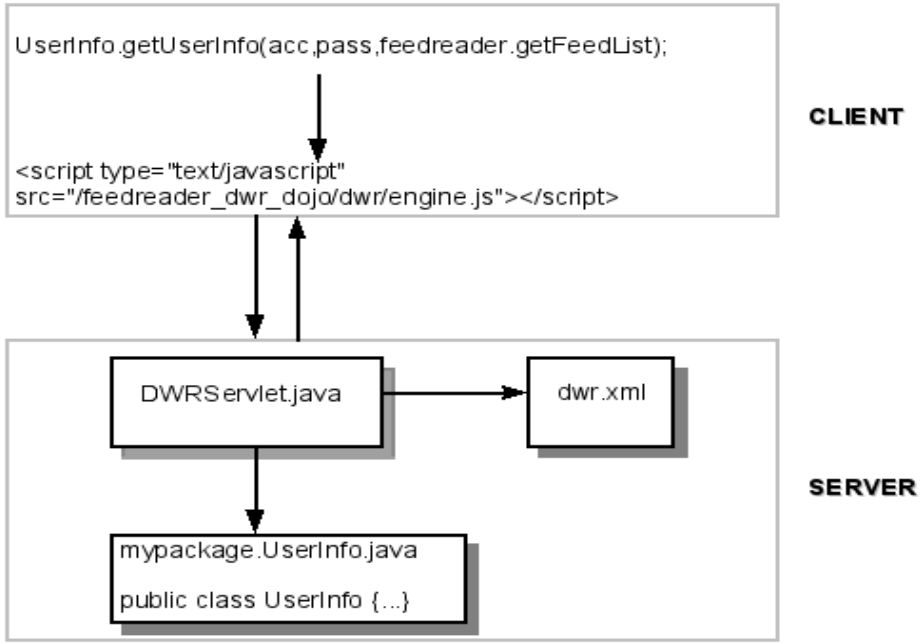
- Simile allo schema di comunicazione generale di una RPC
  - Utilizzo di stub e skeleton.
- E' possibile usare sintassi di tipo RMI all'interno del codice Javascript lato client.
- Il framework genera il client stub, che è codice Javascript
- Runtime lato server.
- Il client stub gestisce la corrispondenza dei parametri e il valore ritorno.



Soluzione proposta 2 : DWR

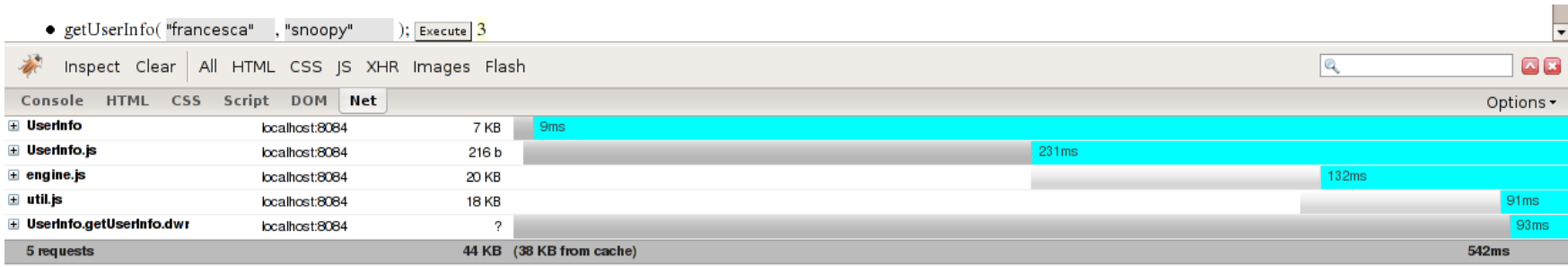
Utilizzo di DWR applicato all'applicazione FeedReader:

- Autenticazione dell'utente tramite la realizzazione della classe Javascript requestUserInfo. Caricamento del feed appropriato.
- Utilizzo delle API per la manipolazione del DOM.



Soluzione proposta 2 : DWR

- Aggiungendo /dwr all'indirizzo dell'applicazione si accede alla pagina di debug.
  - Problema: esposizione dei metodi.
  - Soluzione: si espone all'esterno solo ciò che veramente si desidera
- Non c'è un carico eccessivo determinato dall'inserimento delle librerie.
- Possibilità di usare classi Java da qualsiasi fonte, senza alcuna modifica.



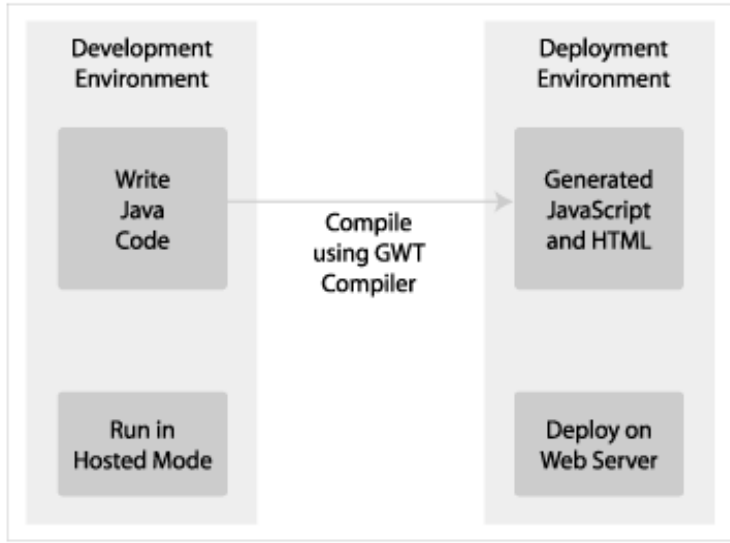
Soluzione proposta 3: GWT

Google Web Toolkit: approccio radicale al problema.

- **GWT permette di sviluppare e debuggare applicazioni Ajax in un ambiente di sviluppo Java**

Due modalità di funzionamento:

1. **Modalità Hosted**: in questa modalità l'applicazione gira come codice Java all'interno della Java Virtual Machine(JVM).
2. **Modalità Web**: l'applicazione gira come Javascript e Html puro compilato a partire dal codice srogente Java attraverso il compilatore Java-to-Javascript.

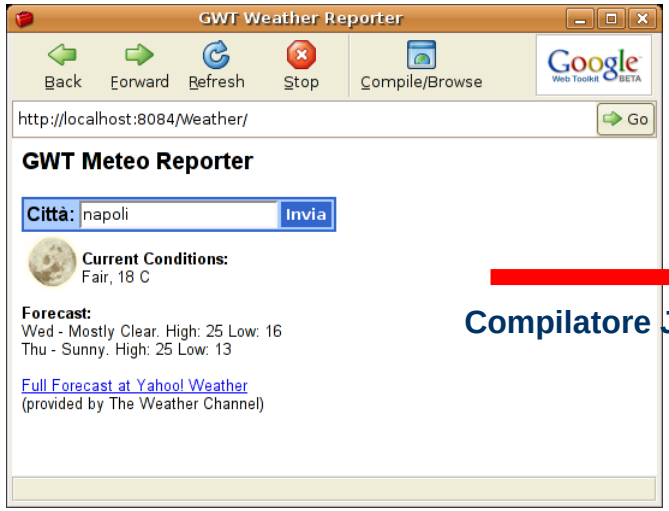


Soluzione proposta 3: GWT

Funzionalità implementate: realizzazione di una pagina GWT per la lettura di un feed Meteo da Yahoo.com:

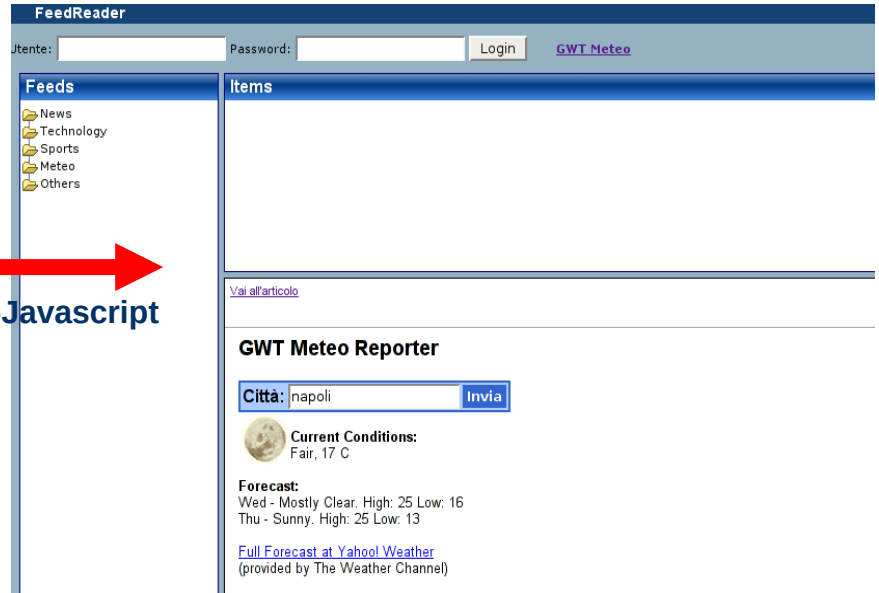
la visualizzazione avviene nel pannello dei messaggi di FeedReader:

Modalità hosted:



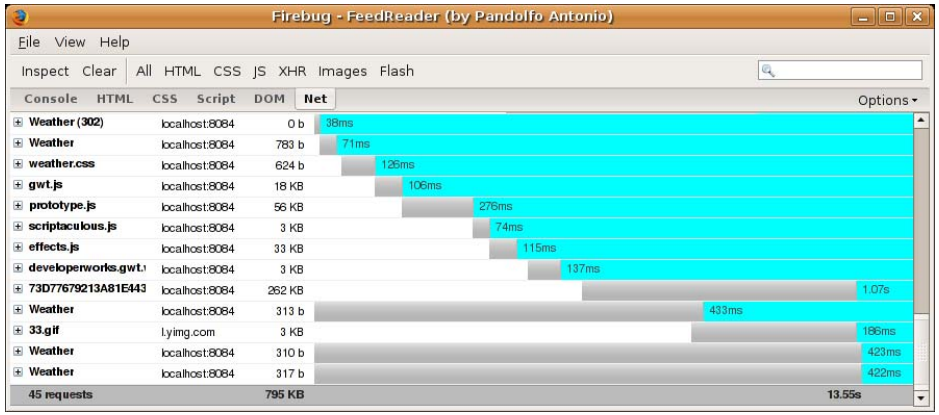
Compilatore Java-to-Javascript

Modalità web:



Soluzione proposta 3: GWT

- Vantaggi:
  - C'è bisogno di conoscere un solo linguaggio di programmazione: JAVA
  - File generati dal compilatore Java-to-Javascript di dimensioni contenute
  - Uso di Javascript relegato a casi particolari e sporadici.



- Svantaggi:
  - Non c'è controllo sul codice generato
  - Difficoltà di debug una volta usciti dal proprio IDE

Conclusioni

- **Dojo e DWR** sono attualmente i framework più utilizzati, essi rappresentano infatti un equo compromesso tra innovazione e legame con le metologie di sviluppo tradizionali.
- **GWT** rappresenta un approccio radicale al problema. L'utilizzo di Java rappresenta un grande vantaggio, ma potrebbe diventare il suo principale limite (scarso controllo sul codice generato).

- Sviluppi possibili:
  - espandere le funzionalità dell'applicazione FeedReader sfruttando ad esempio effetti avanzati come il drag and drop.
  - realizzare l'applicazione FeedReader utilizzando esclusivamente Java.