

Chinosi Michele
mat. 592329

Corso di laurea quinquennale in Informatica (F28)
Facoltà di Scienze MM.FF.NN.
Università degli Studi di Milano

LA SECONDA RELEASE DI VIRTUOSE BASATA SU DATABASE XML

- Studio del database XML -

Riassunto del lavoro di tesi

Anno accademico 2003/2004

Relatore:

Prof. ssa De Cindio Fiorella

Correlatori:

**Benini Marco, Università dell'Insubria, Varese
Sonnante Leonardo, Fondazione RCM, Milano**

L'idea del presente lavoro di tesi è nata dalla necessità di aggiornamento del motore open-source per comunità virtuali Virtuose. Con la precedente release, basata su database PostgreSQL e pagine scritte in PHP, si sono manifestate alcune problematiche legate all'implementazione e alle prestazioni.

Nonostante la solidità e le features offerte dal database PostgreSQL, tuttavia esso non soddisfa alcuni dei criteri in base ai quali è stata concepita l'applicazione Virtuose, come la possibilità di personalizzare agevolmente la struttura del repository da parte degli utenti. Per quanto concerne invece il codice PHP esiste un grosso problema di backward compatibility nel passaggio da una versione del linguaggio alla successiva. Infine, come analizzato nella prima parte della tesi, si evidenzia una debolezza notevole della release dal punto di vista prestazionale nel caso di esecuzione di query particolarmente complesse.

Viste le premesse appena esposte e i suggerimenti forniti da un precedente lavoro di tesi, con la presente ricerca si è cercato di trovare soluzioni alternative che rispettassero i vincoli di stabilità, standardizzazione, universalità, personalizzazione, semplicità e velocità richiesti dalla definizione dell'applicazione Virtuose.

Dal punto di vista implementativo la nuova release del motore di Virtuose si basa su un database costituito mediante il meta-linguaggio XML, così come viene definito nelle specifiche del W3C e da un file XSchema associato contenente la definizione dei tipi di dato presenti nell'archivio. Tale documento permette di memorizzare i dati utilizzando la struttura predefinita del database ma lasciando la possibilità di personalizzazione della struttura e dei contenuti grazie alla presenza di tag di tipo any-Type. E' stato quindi creato un parser in Java per poter validare il documento XML in base alle specifiche ufficiali e in base al documento XSchema (XSD) associato, dando la possibilità di validare direttamente anche il file XSD in base alle specifiche ufficiali del W3C. Le funzioni di parsing si basano sulle classi costituenti il parser stream-based SAX.

Il file XML contenente la struttura e i dati dell'applicazione viene gestito utilizzando il framework di Apache Group Xindice, un database XML nativo

progettato per la gestione di basi di dati costituite da collezioni di documenti XML. Questa soluzione, come evidenziato nel capitolo dedicato alla scelta del repository, risulta l'approccio ottimale per la costituzione e la gestione del motore di Virtuose, grazie anche a un sistema di indicizzazione e di auto-linking studiati al fine di velocizzare l'esecuzione di query particolarmente strutturate. Un insieme di classi in Java permette di scomporre il documento XML principale creando un insieme di collezioni di documenti XML e di file XSchema associati, facilmente importabili all'interno di Xindice.

Per la gestione della struttura e dei dati contenuti nel repository vengono utilizzate classi Java che implementano un insieme di funzioni che permettono l'esecuzione di query Xpath, per la selezione di singoli o gruppi di nodi dalle collezioni di documenti XML, e Xupdate, per l'aggiornamento dei documenti XML.

L'intera architettura del repository con i dati in essa contenuti è stata poi inserita tramite un'operazione di deployment all'interno del web-server Tomcat, sempre di Apache Group, come descritto nella parte di implementazione della tesi. All'interno del server è stato eseguito il deployment anche del framework Jetspeed, un Enterprise Information Portal che potrebbe costituire il web-tier della nuova release di Virtuose. Le classi Java create in precedenza sono state incluse nel path di Jetspeed per permetterne il riutilizzo tramite pagine web scritte utilizzando il linguaggio JSP e le tag-library di Jetspeed. L'architettura a portlet offerta da Jetspeed permette di definire nuove portlet che integrino le pagine JSP a loro volta richiamanti le classi definite in precedenza, permettendo l'accesso al database Xindice e l'esecuzione delle query.

I test prestazionali condotti, analizzati nella parte conclusiva della tesi, unitamente alle informazioni rilevate dalle documentazioni ufficiali e dalle mailing-list, hanno evidenziato un incremento notevole delle prestazioni dell'intera applicazione, mantenendo i vincoli imposti di stabilità e standardizzazione. La semplicità con la quale il database Xindice, il web-server Tomcat e l'EIP Jetspeed possono essere integrati fornisce una soluzione ottimale per lo sviluppo di applicazioni basate sull'architettura proposta.