Università degli Studi della Calabria

Dipartimento di Matematica

Dottorato di Ricerca in Matematica ed Informatica

XX Ciclo

Settore Disciplinare INF/01 INFORMATICA

Tesi di Dottorato

$\mathbf{DLV}^{DB}$
An ASP System for Data Intensive Applications

Claudio Panetta

Supervisori

Prof. Nicola Leone

Dott. Giorgio Terracina

Coordinatore

Prof. Nicola Leone

Anno Accademico 2007 - 2008

DLV^{DB}

An ASP System for Data Intensive Applications

Ph.D. Thesis

Claudio Panetta

Sommario

La rapida crescita di sistemi informatici derivanti dalle diverse applicazioni cui Internet si presta, ha rapidamente aumentato la quantità di dati e di informazioni disponibili per l'elaborazione. In particolare, l'affermarsi del commercio elettronico, il diffondersi di sistemi per l'e-government delle pubbliche amministrazioni, l'ormai avviato processo di digitalizzazioni degli archivi e dei documenti in essi contenuti, la disponibilità di database medici sempre più completi e ricchi di informazioni e, più in generale, il sempre maggiore utilizzo dei sistemi informatici per la gestione strutturata di grandi quantità di dati hanno evidenziato l'urgenza di sviluppare nuove tecnologie che consentano di elaborare automaticamente ed efficientemente la quantità di dati derivante da questi settori emergenti.

Uno degli utilizzi principali dei sistemi di basi di dati (DBMS) consiste nella memorizzazione e nel recupero efficiente di grandi quantità di dati. L'elaborazione di tali informazioni, specialmente quella finalizzata all'estrazione di nuova conoscenza, è ormai riconosciuta come una tematica di ricerca di fondamentale importanza sia nell'ambito delle basi di dati, sia nell'ambito della ricerca industriale, in quanto offre grandi opportunità di sviluppo. In tale scenario, applicazioni come "Data Mining", "Data Werehousing" e "Online Analytical Processing (OLAP)" hanno ulteriormente evidenziato la necessità di sviluppare sistemi di basi di dati che supportino linguaggi maggiormente espressivi, in grado di consentire elaborazioni sempre più raffinate delle informazioni contenute nei Database. Il complesso di tali esigenze ha portato alla definizione di diverse estensioni per i modelli di rappresentazione dei dati (Modelli Relazionali basati sul concetto degli Oggetti), nonché alla definizione di nuovi costrutti sintattici (ricorsione e costrutti OLAP), ed all'estenzione dei DBMS (DataBase Management Systems) con linguaggi di programmazione di alto livello, basati su UDF (User Defined Functions).

Purtroppo, però anche i migliori sistemi di basi di dati attualmente in commercio non sono sufficientemente potenti e generali da poter essere efficacemente utilizzati per risolvere molte delle emergenti applicazioni. In generale, gli attuali DBMS non contengono i meccanismi di ragionamento necessari per estrarre conoscenza complessa dai dati disponibili. Tali meccanismi, dovrebbero essere in grado sia di gestire grandi quantità di informazioni, sia di realizzare sofisticati processi di inferenza sui dati per trarne nuove conclusioni.

Le capacità di ragionamento necessarie a tale scopo possono essere fornite dai sistemi basati su linguaggi logici. La Programmazione Logica Disgiuntiva (DLP) è un formalismo che consente di rappresentare, in maniera semplice e naturale, forme di ragionamento non monotono, planning, problemi diagnostici e, più in generale, problemi di elevata complessità computazionale. In DLP, un programma è una collezione di regole logiche in cui è consentito l'uso della disgiunzione nella testa delle regole e la negazione nel corpo. Una delle possibili semantiche per tali programmi è basata sulla nozione di modello stabile (*answer set*). Ad ogni programma viene associato un insieme di answer set, ognuno corrispondente ad una possibile visione del dominio modellato.

La DLP sotto tale semantica viene comunemente riferita con il termine di *Answer Set Programming* (ASP). Il recente sviluppo di efficienti sistemi basati sulla programmazione logica come DLV [80], Smodels [101], XSB [114], ASSAT [84, 86], Cmodels [62, 61], CLASP [56], etc., ha rinnovato l'interesse nei campi del ragionamento non-monotono e della programmazione logica dichiarativa per la risoluzione di molti problemi in differenti aree applicative. Conseguentemente, tali sistemi possono fornire le funzionalità di inferenza e ragionamento richieste dalle nuove aree di applicazione che interessano i sistemi di basi di dati.

Tuttavia, i sistemi basati sulla programmazione logica presentano notevoli limitazioni nella gestione di grandi quantità di dati non essendo dotati dell'opportuna tecnologia per rendere efficiente la loro gestione poiché eseguono le loro elaborazioni facendo uso di strutture dati gestite direttamente in memoria centrale. Inoltre, la maggior parte delle applicazioni di interesse comune coinvolge grandi moli di dati su cui applicare complessi algoritmi di inferenza logica difficilmente elaborabili sia dai sistemi di programmazione logica, sia dai tradizionali database.

Queste considerazioni mettono in evidenza la necessità di tecniche efficienti ed efficaci che combinino le qualità dei sistemi di inferenza logica con quelle dei sistemi di gestione delle basi di dati. In letteratura, le proposte di soluzione a tale problema sono culminate nei Sistemi di Basi di Dati Deduttive (DDS) [25, 52, 23, 63], che combinano le due realtà dei sistemi logici e dei DBMS. In pratica, i DDS sono il risultato di una serie di tentativi di adattare i sistemi logici, che hanno una visione del mondo basata su pochi dati, ad applicazioni su grandi moli di dati attraverso interazioni intelligenti con le basi di dati. In particolare, i DDS sono forme avanzate di DBMS i cui linguaggi di interrogazione, basati sulla logica, sono molto espressivi. I DDS non memorizzano solo le informazioni esplicite in un database relazionale, ma memorizzano anche regole che consentono inferenze deduttive sui dati memorizzati. L'uso congiunto di tecniche sviluppate nell'ambito delle basi di dati relazionali con quelle della programmazione logica dichiarativa, consente in linea di principio ai DDS di realizzare ragionamenti complessi su grandi quantità di dati.

Tuttavia, nonostante le loro potenzialità lo sviluppo di sistemi DDS a livello industriale non ha ricevuto molta attenzione. Ciò principalmente è stato dovuto al fatto che è estremamente complesso ottenere sistemi particolarmente efficienti ed efficaci; infatti, le attuali implementazioni di DDS sono basate su due approcci estremi: uno basato sul miglioramento dell'elaborazione dei dati da parte dei sistemi logici, l'altro basato sull'aggiunta di capacità di ragionamento ai DBMS (ad esempio tramite l'uso di SQL99, o di funzioni esterne). Entrambi tali approcci presentano limitazioni importanti. In particolare, i DDS basati sulla logica possono gestire una quantità limitata di dati, dal momento che, gli attuali sistemi logici eseguono i loro ragionamenti direttamente in memoria centrale; inoltre, essi forniscono interoperabilità limitate con DBMS esterni. Al contrario, i DDS basati sui database offrono funzionalità avanzate di gestione dei dati, ma scarse capacità di ragionamento (sia a causa della poca espressività dei linguaggi di interrogazione, sia a causa di problemi di efficienza).

Riassumendo, possiamo affermare che:

- Gli attuali sistemi di basi di dati implementano moduli sufficientemente robusti e flessibili capaci di gestire grandi quantità di dati, ma non possiedono un linguaggio sufficientemente espressivo da consentire ragionamenti complessi su questi dati.
- I sistemi basati sulla programmazione logica, possiedono elevate capacità di ragionamento e sono in grado di modellare e risolvere con facilità problemi di elevata complessità ma presentano notevoli limitazioni nella gestione di grandi quantità di dati poiché eseguono le loro elaborazioni facendo uso di strutture dati gestite direttamente in memoria centrale.

- I sistemi di basi di dati deduttive consentono di gestire i dati memorizzati su DBMS, ma, dal momento che, eseguono i loro ragionamenti direttamente in memoria centrale, possono gestire una quantità limitata di dati;

Dalle precedenti osservazioni, si evidenzia la necessità di realizzare applicazioni che combinino il potere espressivo dei sistemi di programmazione logica con l'efficiente gestione dei dati tipica dei database.

Il contributo di questa tesi si colloca nell'area della ricerca sulle basi di dati deduttive con l'obiettivo di colmare il divario esistente tra sistemi logici e DBMS. In questa tesi viene descritto un nuovo sistema, DLV^{DB}, che ha la caratteristica di possedere le capacità di elaborazione dati desiderabili da un DDS ma di supportare anche le funzionalità di ragionamento più avanzate dei sistemi basati sulla programmazione logica disgiuntiva.

DLV^{DB} è stato progettato come estensione del sistema DLV e combina l'esperienza maturata nell'ambito del progetto DLV nell'ottimizzare programmi logici con le avanzate capacità di gestione dei dati implementate nei DBMS esistenti. Ciò consente di applicare tale sistema in ambiti che necessitano sia di valutare programmi complessi, sia di lavorare su grandi quantità di dati. DLV^{DB} è in grado di fornire, così sostanziali miglioramenti sia nelle prestazioni relative alla valutazione dei programmi logici, sia nella facilità di gestione dei dati di input e di output possibilmente distribuiti su più database. L'interazione con le basi di dati è realizzata per mezzo di connessioni ODBC che consentono di gestire in modo piuttosto semplice dati distribuiti su vari database in rete. DLV^{DB} consente di applicare diverse tecniche di ottimizzazione sviluppate sia nell'ambito dei sistemi logici, come ad esempio i magic set, sia nell'ambito della gestione delle basi di dati, come ad esempio tecniche di join ordering, inoltre sono stati integrati nel sistema i predicati per l'aggregazione di DLV (count, min, max, avg, sum) che avvicinano il linguaggio alle potenzialità di SQL, ma anche la possibilità di integrare nel programma logico, per natura dichiarativo, chiamate a funzioni esterne sviluppate con tecniche procedurali; ciò rende possibile integrare aspetti dichiarativi ed aspetti procedurali di un problema in un'unica framework. Infine, per consentire la gestione di tipi di dati con strutture ricorsive (es. XML) si è introdotta la possibilità di gestire liste di elementi, eventualmente innestate, nel programma logico.

Inoltre in questa tesi viene presentata l'attività di analisi di tipo sperimentale effettuata al fine di valutare le prestazioni di DLV^{DB}, soprattutto in riferimento a velocità di esecuzione di query e quantità di dati gestibili. Questi test hanno dimostrato come il sistema apporta numerosi vantaggi rispetto ai sistemi esistenti, sia in termini di tempi di esecuzione delle query, sia in termini di quantità di dati che esso riesce a gestire contemporaneamente.

In sintesi, i contributi di questo lavoro possono essere riassunti come segue:

- Sviluppo di un sistema in grado di fondere il potere espressivo dei sistemi ASP con l'efficiente gestione dei dati offerta dagli attuali Database;
- Sviluppo di una strategia di valutazione dei programmi logici in grado di minimizzare l'utilizzo della memoria centrale massimizzando l'utilizzo delle tecnologie implementate dai DBMS;
- Estensione del linguaggio DLP mediante l'introduzione di chiamate a funzioni esterne e il supporto a tipi di dati con strutture ricorsive come le liste;
- Realizzazione di un'analisi comparativa tra le prestazioni offerte da DLV^{DB} e le prestazioni dei sistemi esistenti.