

RELAZIONE

Domanda 4

Come illustrato a lezione, SimpleDB utilizza la strategia naïf per il rimpiazzamento dei buffer.

Implementare almeno una delle altre strategie (FIFO, LRU, clock) e mostrare che si ottengono così significativi benefici.

Notare anche che la versione scaricabile di SimpleDB utilizza pochissimi buffer(8), e che per migliorare le prestazioni è necessario aumentarli; valutare quale può essere un numero adeguato per ottenere benefici.

Individuare opportune modalità per mostrare le differenze di prestazioni. Si suggerisce di lavorare con un programma client, documentando il numero di letture di blocchi effettuate. Allo scopo, si può procedere come segue:

- Modificare la classe FileMgr in modo che supporti statistiche sul numero di blocchi letti (aggiungere uno o più metodi), con riferimento a ciascuno dei file usati da simpleDB (così si potrà capire dove è l'overhead)

- Modificare il metodo commit della classe RemoteConnectionImplementation (in simpledb.remote) in modo che stampi queste statistiche (si deve accedere al FileMgr che può essere ottenuto con il metodo SimpleDB.fileMgr, nel package simpledb.server). Si possono vedere risultati interessanti anche eseguendo attraverso il client SQLInterpreter una semplice interrogazione, quale `SELECT SName FROM Student` sulla base di dati studentdb fornita con il sistema.

Consegnare le classi modificate e una breve relazione che illustri le modifiche e descriva i test effettuati, mostrandone i benefici ottenuti.

Si è proceduto seguendo i suggerimenti:

- creare una classe per memorizzare le statistiche d'aiuto:

```
simpledb.stats.BasicFileStats
```

- private int blockRead;
- private int blockWritten;
- metodi getter e setter, incrementer

-

- aggiungere in simpledb.buffer.Buffer

- la variabile private long lastUnpinTimestamp per memorizzare il timestamp dell'unpin

- nel costruttore

```
setLastUnpinTimestamp(Calendar.getInstance().getTimeInMillis());
```

- nel metodo unpin

```
setLastUnpinTimestamp(Calendar.getInstance().getTimeInMillis());
```

- metodi getter e setter per il lastUnpinTimestamp.

- aggiungere in simpledb.buffer.BasicBufferMgr:

- private int lastReplacedPageIndex variabile per la strategia che contiene l'ultimo indice di pagina usato

- metodo private Buffer chooseUnpinnedBuffer_Clock()

- metodo private Buffer chooseUnpinnedBuffer_LRU()

- metodo private int getBufferSuccessorIndex(int pageIndex) chiamato da Clock

-

- modificare simpledb.file.FileMgr:

- private Map<String, BasicFileStats> blockStatsFile variabile per le statistiche

- metodi per aggiornare la lettura dei blocchi

```
private void updateReadBlockStats(Block blk, ByteBuffer bb)
```

```
private void updateWriteBlockStats(Block blk, ByteBuffer bb)
```

- nel metodo getFile aggiornare le statistiche:

```
if ( !this.blockStatsFile.containsKey(filename) ) {
```

```
this.blockStatsFile.put(filename, new BasicFileStats());
```

- metodi per accedere alle statistiche:

```
public final Map<String, BasicFileStats> getMapStats()
```

```
public final void resetMapStats()
```

- modificare `simpledb.remote.RemoteConnectionImpl` :
 - Aggiungere un metodo `private static void printBlockStats(String fileName, BasicFileStats fileStats)` e un metodo `private static void printAllBlockStats()`
 - Nel costruttore `printAllBlockStats();`
 - `SimpleDB.fileMgr().resetMapStats();`
 - Nel metodo `commit` chiamare il metodo `printAllBlockStats();`

I test effettuati si trovano tutti nel file `BasicFileStats.pdf`. Dai test fatti vediamo che la strategia NAIF è inefficiente anche se aumentiamo il numero di buffer non otteniamo nessun beneficio mentre con una delle altre strategie si ottengono benefici con l'aumentare dei buffer. Le statistiche ci suggeriscono che il miglior numero di buffer in questo caso utilizzando la strategia LRU è 15, se aumentiamo ulteriormente non otteniamo benefici infatti se inizializzo 1000 buffer il risultato di letture è cambiato solo di uno. Mentre con la strategia clock i risultati ci mostrano che con 14 buffer si ha circa il doppio delle letture rispetto a LRU e solo con 32 buffer riesce a fare più o meno le stesse letture della strategia LRU aumento ancora i buffer non si ottengono benefici.