

SISTEMI OPERATIVI

Sistemi Distribuiti
Computazione Distribuita

Lezione 5 – Coordinamento distribuito tra processi (parte 2)

Vincenzo Piuri

Università degli Studi di Milano

Atomicità (1)

- Coordinatore delle transazioni in ciascuna macchina
 - Iniziare la transazione
 - Dividere la transazione in sotto-transazioni
 - Distribuzione delle sotto-transazioni sulle macchine adatte
 - Coordinamento della fine della transazione



Protocollo di commit a due fasi

- Transazione inizia su macchina S_i
- Quando tutti i siti comunicano a S_i che è stata completata la porzione di T ad essi assegnata si attiva il protocollo a due fasi per definire il commit della transazione
- Fase 1
 - C_i aggiunge <prepare T> e manda messaggio "prepare T" a tutte le macchine coinvolte
 - Quando tale messaggio è ricevuto da una macchina S_j , C_j decide se commit sulla macchina è possibile
 - Se non possibile, aggiunge <no T> e risponde a C_j con abort
 - Se possibile, aggiunge <ready T> e risponde a C_j con "ready T"

Protocollo di commit a due fasi

- Fase 2
 - Quando C_i riceve le risposte oppure dopo che è trascorso time-out, C_i stabilisce se può chiudere la transazione
 - T è chiusa con commit se sono ricevuti `<ready T>` da tutte le macchine, aggiungendo `<commit T>` al log
 - Altrimenti T è abortita aggiungendo `<abort T>`
 - Il risultato è trasmesso a tutte le macchine
 - Quando un coordinatore locale riceve “commit T ” o “abort T ”, registra `<commit T>` o `<abort T>` rispettivamente

Protocollo di commit a due fasi

- Tolleranza ai guasti delle macchine
- Tolleranza ai guasti della rete

Ambiente distribuito

- Gestore delle transazioni
 - Transazioni locali
 - Transazioni globali
- File di log
- Protocolli
 - bloccanti
 - non bloccanti

Protocolli bloccanti (1)

- Realizzazione del gestore del blocco in ambiente distribuito
- Replicazione dei dati
- Modalità di gestione del blocco
condiviso o esclusivo

Protocolli bloccanti (2)

Coordinatore centralizzato dei lock

- Dati non replicati
- Unico responsabile centralizzato dei lock
- Realizzazione semplice:
richiesta di lock mediante scambio di messaggi
(2 messaggi)
- Gestione stallo: complicata
- Prestazioni: limiti dovuti a coordinatore unico
- Tolleranza ai guasti: critica dovuta
alla centralizzazione

Protocolli bloccanti (3)

Coordinatori multipli dei lock

- Dati non replicati
- Responsabile locale dei lock
- Realizzazione semplice:
richiesta di lock mediante scambio di messaggi
(2 messaggi)
- Gestione stallo: complicata

Protocolli bloccanti (4)

Coordinatore dei lock a maggioranza

- Dati replicati
- Responsabile dei lock per ogni sito
- Richiesta di almeno $n/2+1$ lock locali;
lock globale ottenuto quanto almeno $n/2+1$ lock locali sono ottenuti
- Realizzazione complicata
 - $2(n/2 + 1)$ messaggi per gestire il blocco
 - $(n/2 + 1)$ messaggi per gestire lo sblocco
- Bisogna modificare gli algoritmi per la gestione dello stallo

Protocolli bloccanti (5)

Protocollo polarizzato

- Dati replicati
- Responsabile dei lock per ogni sito
- Blocchi condivisi: richiesta locale
- Blocchi esclusivi: richiesta globale come in lock a maggioranza
- Minore sovraccarico nella lettura rispetto ai protocolli di lock a maggioranza
- Ulteriore sovraccarico in scrittura
- Difficile gestire lo stallo

Compito

- Gestione del coordinamento tra processi in ambiente distribuito
 - Mutua esclusione
 - Rilevazione degli stalli
 - Sostituzione token persi
 - Controllo input/output

Algoritmi di elezione

- Processo con identificatore di priorità più alto
- Algoritmo del bullo
- Algoritmo dell'anello

Algoritmo del bullo

P cerca di eleggere se stesso come coordinatore.

Quando P si accorge che il coordinatore non funziona

- manda un messaggio di inizio elezione ai processi con priorità più alta e attende risposta
- se nessuna risposta entro timeout T, P elegge se stesso come coordinatore e informa tutti i processi che ha assunto tale ruolo
- se P riceve una risposta, allora attende l'identificatore del nuovo coordinatore
- se P non riceve l'identificatore del nuovo coordinatore, deve far ripartire l'elezione

Coordinatore (4)

Algoritmo dell'anello

- Collegamento logico ad anello unidirezionale
- Lista attiva

Quando P si accorge che il coordinatore non funziona

- genera una nuova lista attiva vuota
- attiva elezione inviando la lista attiva al processo successivo nell'anello

Se P riceve un messaggio di elezione

- se è primo messaggio ricevuto o inviato, crea una lista attiva includendo se stesso e il processo che lo precede nell'anello e invia un messaggio di elezione indicando se stesso e il processo predecessore
- se il messaggio ricevuto non contiene P, aggiunge il predecessore alla propria lista attiva ed inoltra il messaggio di elezione al processo successivo
- se il messaggio ricevuto contiene P, la lista attiva contiene tutti i processi: P può identificare il coordinatore nella lista come il processo con priorità più elevata

In sintesi

- Ordinamento degli eventi
- Tecniche per la sincronizzazione
 - Mutua esclusione
 - Atomicità
 - Gestione concorrenza