

Ottimizzazione scheduler parallelo

Guastella Marco

3 maggio 2009

Indice

1	Ottimizzazione scheduler parallelo	4
1.1	L'ottimizzazione	4
1.2	Bilancio sugli eventi parallelizzabili	5
1.3	Impatto temporale della simulazione	6

Elenco delle figure

1.1	Eventi eseguiti dai Local Process nelle simulazioni	5
-----	---	---

Elenco delle tabelle

1.1	Percentuale remote positive nel modello di esempio	6
1.2	Lunghezza media code nel modello di esempio	6

Capitolo 1

Ottimizzazione scheduler parallelo

Per la comprensione di questo documento bisogna leggere il *capitolo 6 di tesi-pdnet.pdf* per capire il funzionamento dello scheduler parallelo di Pdnet. Lo scopo di questo documento è quello di applicare un'ottimizzazione allo scheduler parallelo centralizzato descritto nel *capitolo 6 di tesi-pdnet.pdf* e analizzarne le performance.

1.1 L'ottimizzazione

Il Local Process Master ad ogni ciclo di routine controlla se il prossimo evento da eseguire si trova nella coda degli eventi da eseguire (Event Code) o nella coda degli eventi eseguiti dagli slave ma non ancora verificati dal master (Conditional Event Code). Se l'evento si trova nella coda degli eventi da eseguire il master assegna ad ogni slave un evento partendo dal secondo evento nella Event Code se invece l'evento si trova nella coda degli eventi eseguiti dagli slave ne verifica la validità.

Analizzando i risultati del *paragrafo 6.4.2 di tesi-pdnet.pdf* si nota che le simulazioni parallele con più di due Local Process non danno risultati soddisfacenti perchè la percentuale di validità degli eventi eseguiti dagli slave è bassa. Il motivo di queste percentuali basse è dato dal fatto che gli eventi assegnati agli slave vengono eseguiti molto in anticipo rispetto alla loro reale esecuzione e la condizione di validità non è quasi mai verificata. L'ottimizzazione applicata allo scheduler parallelo serve a risolvere questo problema. L'ottimizzazione applicata allo scheduler parallelo riguarda l'assegnazione degli eventi del Master agli Slave. Il master assegna agli slave solo gli eventi della Event Code che devono essere eseguiti prima dell'evento in testa della Conditional Event Code. Se la Conditional Event Code è vuota si comporta come l'algoritmo senza ottimizzazione. Questa ottimizzazione non fa altro

che aggiungere un intervallo di sincronizzazione conservativo ad una tecnica ottimistica.

1.2 Bilancio sugli eventi parallelizzabili

Le simulazioni effettuate sono una seriale e sette simulazioni parallele con (2,3...,7 LP) utilizzando il modello di esempio utilizzato per le simulazioni precedenti (*paragrafo 6.4.1 tesi-pdnet.pdf*). Il numero di eventi eseguiti dai Local Process nelle simulazioni è descritto in *figura 1.1*. Dall'analisi della fi-

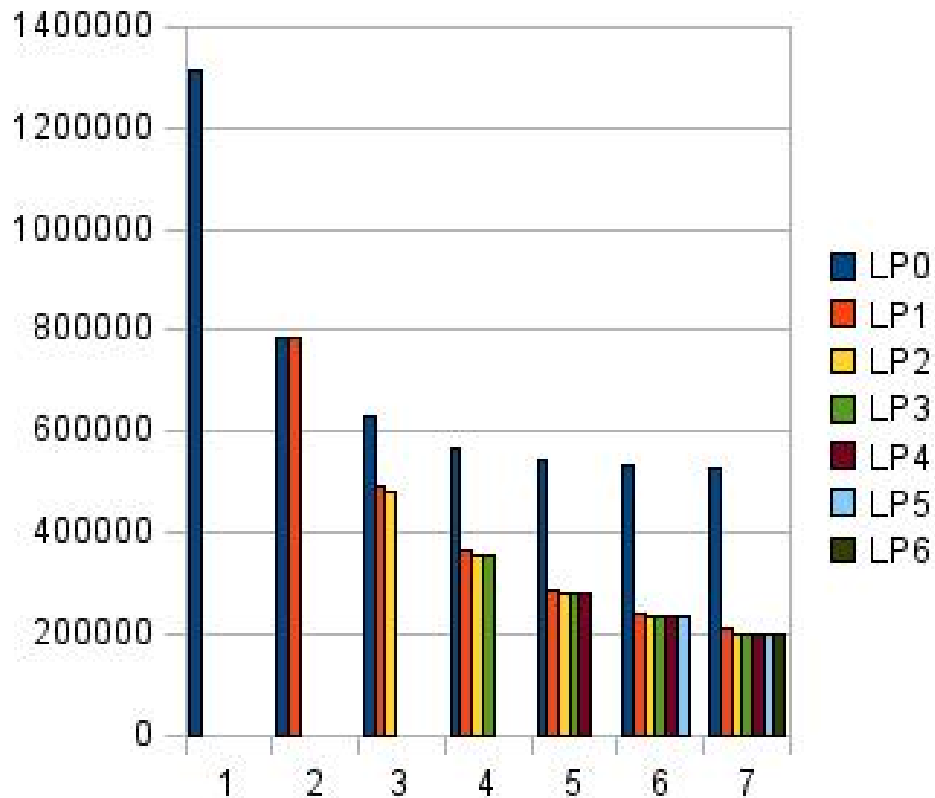


Figura 1.1: Eventi eseguiti dai Local Process nelle simulazioni

gura si nota che i risultati della simulazione con 2 Local Process sono identici alle simulazioni con lo scheduler parallelo senza ottimizzazione. Nelle simulazioni con più di due Local Process gli LP slave eseguono quasi lo stesso carico di lavoro e non esistono LP inutilizzati. Gli eventi eseguiti dal master nelle simulazioni con più di due Local Process sono sempre di un numero molto

superiore agli eventi eseguiti da un singolo slave ma diminuiscono aumentando il numero di Local Process. Con più di 7 Local Process il numero di eventi eseguiti dal master aumentano. Questo risultato è importante perché con eventi con tempi di esecuzione alti, tanto da trascurare i tempi di latenza per lo scambio dei messaggi e i tempi di controllo che sono quasi nulli, si ha un guadagno nella diminuzione del numero di eventi eseguiti dal master. Le percentuali degli eventi eseguiti dagli slave che risultano validi sono descritti nella *tabella 1.1*.

numero LP	% remote positive
2	67.7%
3	70%
4	69.8%
5	68.35%
6	66.7%
7	65.49%

Tabella 1.1: Percentuale remote positive nel modello di esempio

I risultati delle remote positive dimostrano che fino a 7 Local process ci sono delle buone percentuali di validità degli eventi eseguiti dagli slave. Un'altro risultato da considerare è quello che descrive la lunghezza media delle code durante la simulazione (1.2). Da questi risultati si nota che gli

numero LP	size EC	size CEC
2	457	1
3	456	2
4	456	2
5	455	3
6	455	3
7	454	4

Tabella 1.2: Lunghezza media code nel modello di esempio

eventi che si trovano nella CEC sono pochi e con questa ottimizzazione si è risolto il problema dell'esecuzione degli eventi molto in anticipo rispetto alla loro reale esecuzione da parte degli slave.

1.3 Impatto temporale della simulazione