ISTITUTO NAZIONALE DI FISICA NUCLEARE

Pisa

INFN/CCR-11/01
26 Agosto 2011

CCR-41/2011/P

INTERVENTI SU KERNEL E MICROCODE PER ADEGUARE IL PROCESSORE AMD 8356 REV. B2 ALL'AMBIENTE GRID

Dario Fabiani, Enrico Mazzoni

[1)]*INFN - Sezione di Pisa, Largo B. Pontecorvo, 3, I-56127 Pisa, Italy*

Abstract

Nel periodo 2009-2010 la Sezione di Pisa ha installato e messo in produzione un cluster da 1024 core composto dal 128 server bi-processori equipaggiati con CPU AMD Opteron 8356 rev. B2, quad core. Subito dopo l'avvio del cluster (sotto GRID) sono stati riscontrati problemi di prestazioni legati al processore. Il presente lavoro descrive la situazione registrata e gli interventi apportati al Kernel Linux ed al Microcode per ovviare al problema

Published by ***SIDS–Pubblicazioni***
Laboratori Nazionali di Frascati

Indice

1 Premessa **3**

2 Diagnosi **4**

3 Cura **6**

4 Effetti collaterali **7**

5 Conclusioni **8**

Riferimenti bibliografici **9**

1 Premessa

Durante il periodo 2009-2010 è stato messo in produzione un cluster da 1024 core composto da server bi-processori equipaggiati con CPU AMD Opteron 8356, in gergo chiamato "Tramontana". Nelle fasi di avvio della macchina si è deciso di verificarne il funzionamento aprendone l'accesso a normali job grid e quindi controllando la funzionalità dei sistemi e le loro prestazioni. In questa prima fase di funzionamento è emerso che alcune VO lamentavano dei tempi di esecuzione dei propri processi molto più elevati sui nuovi nodi rispetto al resto delle macchine del sito e comunque non paragonabili con macchine equipaggiate dallo stesso tipo di processori. Abbiamo quindi deciso di iniziare una campagna di test approfonditi per capire da che cosa potesse essere causato questo comportamento, i test si sono articolati su due fronti:

- verifica di configurazioni diverse dei sistemi del cluster, modificando quantità e disposizione della memoria o i parametri del BIOS;
- test comparativi fra sistemi equipaggiati con altri processori sempre AMD ma di generazioni diverse.

Per eseguire i test si è deciso di usare il benchmark HEP-Spec2006 (HS06) [1] dato che esso ben rappresenta i tipici job che girano sulle nostre macchine e comunque fornisce numeri che possono essere paragonati con quanto si trova riportato, per esempio, sul sito di spec.org [2] da cui il test è derivato.

I risultati dei test eseguiti con configurazioni diverse non hanno evidenziato modifiche sostanziali delle prestazioni, di fatto escludendo un problema della configurazione hardware o software di singoli componenti dei nodi, questi hanno prestazioni scarse nel loro complesso. I risultati dei test eseguiti su sistemi diversi sono riportati in tabella 1, in cui per ciascuno dei sistemi provati sono riportati: il tipo di processore con fra parentesi i numeri rappresentanti "CPU family" "model" e "stepping", il numero di core totali del sistema, il risultato del test HEP-Spec e il risultato normalizzato per il numero di core ed il clock. Per le caratteristiche dei vari processori si veda [3].

Dai risultati di tabella 1 emergono chiaramente due conclusioni:

1. i processori del tipo (16,2,2) hanno un comportamento diverso dagli altri;
2. il comportamento di questi processori è influenzato dal tipo di piattaforma su cui sono montati;

alla luce di queste due conclusioni preliminari si è deciso di creare un sistema di test basato su una macchina Acer Altos R520 su cui provare versioni diverse di processori

Nodo	Modello	CPU	n. core	HS	HS norm.
csn4wn46	Acer Altos R520	8356 (16,2,2)	8	30.54	1.660
csn4wn28	Acer Altos R520	8356 (16,2,2)	8	31.88	1.733
thprinwn1	DELL SC1465	2380 (16,4,2)	8	77.74	3.887
gridwn182	SUN X4600	8536 (16,2,2)	32	100.98	1.372
gridwn154	Supermicro	8356 (16,2,2)	8	62.23	3.382
gridwn170	GigaByte	2356 (16,2,3)	8	51.43	2.795
gridwn183a	Supermicro	2358SE (16,2,3)	8	68.04	3.562

Tabella 1: Test comparativi di vari sistemi equipaggiati con processori AMD Opteron

CPU	n. core	HS	HS norm.
2360SE (16,2,3)	8	65.25	3.262
8356 (16,2,2)	8	33.08	1.798

Tabella 2: Test comparativi di vari processori AMD Opteron su piattaforma Altos R520

AMD in modo da fissare tutte le condizioni al contorno (scheda madre, BIOS, RAM ecc...) ed evidenziare i soli effetti della CPU. I risultati di questa seconda campagna di test sono riportati in tabella 2, i processori usati in questo caso sono entrambi basati su core di tipo Barcellona ma di revisioni diverse, B2 nel caso 8356 e B3 nel caso del 2360.

Dall'insieme dei risultati di queste prove si conclude che il problema riscontrato è specifico dei processori 8356 basati su core Barcellona B2 per cui ci siamo messi alla ricerca di possibili anomalie di questa tipologia di processori che possano spiegare quanto misurato.

2 Diagnosi

Dalle ricerche effettuate è emerso che i processori AMD Quad Core Opteron 8356 B2, che equipaggiano il cluster Tramontana soffrono di un bug (AMD Erratum 298 [4]) che ne compromette il funzionamento. Secondo quanto riportato dalla casa madre il difetto si manifesta con sporadici crash della macchina, specialmente in condizioni di carico elevato sulla CPU, quindi su un tipico worker node, tale evento è praticamente certo. Ovviamente la soluzione più semplice, ossia sostituire i processori difettosi con altri, non è praticabile nel nostro caso avendo noi circa 150 processori di questo tipo in produzione. Inoltre, trattandosi di un vero e proprio difetto hardware, non si può correggere il problema con un semplice aggiornamento del microcodice al boot del sistema.

La soluzione proposta da AMD è un aggiornamento del BIOS delle macchine che in pratica durante il boot, se riconosce una CPU "buggata", disabilita il componente di-

Test	2360 (B3)	8356 (B2)	B2/B3
471.omnetpp	5.170	1.951	0.377
473.astar	6.900	1.744	0.253
483.xalancbmk	4.401	0.700	0.159
444.namd	12.188	11.150	0.915
447.dealII	14.088	12.338	0.876
450.soplex	5.678	4.440	0.782
453.povray	15.800	14.488	0.917

Tabella 3: Dettaglio dei test che compongono un run HEP-Spec per processori B2 e B3

fettoso (il TLB, Translation Lookaside Buffer, per maggiori dettagli [5]). Occorre che il produttore abbia rilasciato un BIOS con tale patch, per quanto emerso dalle nostre ricerche tale patch è stata adottata da tutti i produttori di hardware, di sicuro nel nostro caso è disponibile. I server così aggiornati effettivamente non soffrono più di crash casuali a scapito però delle prestazioni. Infatti il non funzionamento del TLB provoca sicuramente una diminuzione della velocità degli accessi alla memoria, specialmente nel caso di frequenti cambi pagina (occorre accedere ogni volta alla Page Table), l'AMD dichiara che "in media" il rallentamento è di "circa il 10%".

Riassumendo, dalle ricerche effettuate su descritte e dalle misure eseguite ed illustrate nel capitolo precedente, si può dire che:

- i processori 8356 in nostro possesso, essendo tutti rev. B2, sono affetti dal problema hardware che va sotto il nome di "AMD erratum 298";
- tutte le nostre macchine, ad eccezione di poche Supermicro, hanno il BIOS con applicata la patch suggerita da AMD il che le rende stabili;
- dai test riportati nelle tabelle 1 e 2 è chiaro che per il nostro tipo di applicazioni l'impatto di tale patch è ben oltre quanto dichiarato da AMD;
- inoltre se analizziamo in dettaglio i vari test che compongono un run HEP-Spec si nota che l'impatto della patch influenza in maniera drammatica alcuni di questi mentre è quasi nullo su altri, si veda la tabella 3.

A questo punto sembra ragionevole imputare le scarse prestazione dei processori alla combinazione processore affetto da bug e patch del BIOS. Tale patch non è una buona soluzione per il nostro ambiente avendo un impatto ben oltre il 10% dichiarato sulle prestazioni dei sistemi, dobbiamo quindi esplorare altre strade per poter risolvere il problema.

Kernel	HS	HS norm.
2.6.16.60-0.21[1]	33.08	1.798
2.6.32.12-0.7[2]	37.04	2.013
2.6.23.17 amdpatch	68.57	3.727

Tabella 4: Risultati test su sistema con 8356 B2 e varie versioni di Kernel senza e con patch AMD

3 Cura

Quindi se da un lato abbiamo la necessità di risolvere l'instabilità dei sistemi derivante dall'erratum 298, dall'altro dobbiamo riuscire a farlo senza perdere troppo in prestazioni del sistema; la soluzione ad entrambe le necessità può essere il Kernel. L'AMD ha postato su x86_64.org una patch per il Kernel vanilla 64bit 2.6.23.17 che dovrebbe risolvere il problema, in breve riabilita il TLB (qualora il BIOS l'avesse disabilitato) ed agisce in maniera tale che le condizioni che causano il crash della macchina non si verifichino mai. In questo caso, l'effetto sulle prestazioni è sicuramente molto minore, in teoria dovrebbe essere anche difficile da misurare. Questa è la soluzione che abbiamo scelto, anche se non esente da difficoltà. Per prima cosa la patch è dichiarata non testata nè tanto meno garantita in alcun modo per l'uso in sistemi di produzione. Inoltre il Kernel sul quale è stata sviluppata non è quello standard nè di Scientific Linux 5, cosa questa che non costituirebbe un grosso problema dato l'utilizzo del meccanismo di chroot che facciamo per i nostri sistemi grid, ma neanche di altre distribuzioni in particolare SUSE Enterprise che noi utilizziamo per i nostri sistemi. Quindi la strada intrapresa è stata quella di utilizzare il Kernel vanilla 2.6.23.17 a cui è stata applicata la patch AMD e compilato per adattarsi sia l'hardware di Tramontana sia all'ambiente SUSE Enterprise utilizzato su queste macchine. Il risultato di questo processo è stato un RPM che è stato installato su tutti i sistemi dotati di processori 8356 B2.

Per verificare che il nuovo Kernel davvero risolvesse il problema sono stati ripetuti i test con HEP-Spec, i risultati si trovano in tabella 4. Come si vede l'utilizzo di Kernel "stock" SLES di generazioni diverse non fornisce particolari benefici, la piccola differenza fra i primi due risultati riportati in tabella è attribuibile alle naturali fluttuazioni del test stesso e al fatto che le versioni di gcc (era quello standard della distribuzione) utilizzate nei due test sono leggermente diverse. L'uso del Kernel a cui è stata applicata la patch AMD invece risolve il problema, l'ultima riga della tabella 4 è da confrontarsi con i dati della tabella 1 da cui è chiaro che il valore ottenuto è in linea con i risultati di processori rev. B3 (processori di tipo (16,2,3) nella tabella 1) o comunque rev. B2 su sistemi senza patch del BIOS (sistema gridwn154 nella tabella 1).

4 Effetti collaterali

Dopo un po' di tempo dall'adozione del nuovo Kernel sulla farm Tramontana è emerso un altro problema: sulle macchine con il Kernel 2.6.23.17-amdpatch, e solo su quelle, il sistema di batch (noi utilizziamo LSF) rimuoveva alcuni job (nell'ordine del 2%) ritenendo che questi avessero superato il limite di cputime fissato per la coda. Analizzando il dettaglio di questi job è emerso però che il RUNTIME al momento della loro rimozione era inferiore del CPUTIME quindi in qualche modo quest'ultimo era sbagliato. Come misura di emergenza abbiamo eliminato il limite sul cputime da LSF, così che i job potessero terminare normalmente.

Questo però non ha eliminato del tutto il problema, i job potevano terminare normalmente ma esaminando le statistiche riportate da LSF si vedeva che il cputime di questi job risultava essere sempre dell'ordine di 22Msec. Questi valori, anche se riportati per pochi job avevano la conseguenza di far alterare le statistiche HLR globali del sito di INFN-PISA. Ad esempio l'efficienza totale del sito diventava molto più del 100%, dell'ordine di molte migliaia.

In un sistema dove vogliamo fare un accounting accurato, in modo che sia possibile sapere nel dettaglio quanto sono utilizzate la macchine e chi le ha utilizzate, tutto ciò non è accettabile.

Indagando il problema si è visto che l'origine era nel Kernel stesso che riportava alcuni valori temporali relativi ai processi in maniera errata, LSF non faceva altro che leggerli. Infatti, i normali tool (es. top) girati su una macchina con job "strano" in esecuzione, davano valori errati per il cputime. Anche la lettura diretta tramite */proc/[pid]/stat* conteneva il fantomatico valore 22Msec per il cputime.

Questo problema secondo noi è indipendente dalla patch AMD, visto che nel Kernel 2.6.23 è stata introdotta una modifica sostanziale dello scheduler, il "Completely Fair Scheduler" con accounting granulare in nanosecondi, completamente diverso dal precedente. Inoltre ci sono segnalazioni dello stesso problema con versioni di Kernel completamente diverse (es. SL4) e senza patch AMD [7]. Può anche darsi che questo problema fosse abbastanza diffuso in quanto non influenza il normale funzionamento ma solo l'accounting dei processi, per molti utilizzi si manifesterebbe solamente in valori sballati visibili (raramente) con utility tipo top. Approfondendo l'analisi, è emerso che il problema si genera del fatto che la routine che calcola il cputime ha la possibilità (raramente) di ritornare un incremento negativo. Questo valore viene poi passato ad una funzione di conversione dei tempi da jiffies a ns che vuole il valore d'ingresso "unsigned", causando

[1] SLES 10.2, distribuzione usata sui sistemi in produzione
[2] SLES 11.1, utilizzata solo per questi test

quindi la creazione del numero “abnorme”. Capito questo problema si è deciso di correggere nuovamente il Kernel ma cercando di modificarlo il meno possibile per evitare di introdurre altri errori. Per questo si è inserito un semplice controllo sul segno del valore dell’incremento in modo da evitare l’insorgere della condizione di errore, con questa nuova versione di Kernel non abbiamo più riscontrato problemi di alcun tipo.

5 Conclusioni

Dalle analisi fatte si è concluso che il problema delle prestazioni non soddisfacenti del cluster Tramontana era da ricondursi ad un bug hardware dei processori che lo equipaggiano. La soluzione standard applicata dai produttori dei server tramite patch al BIOS ha un impatto intollerabile sulle prestazioni dei sistemi e quindi l’unica strada percorribile passa attraverso l’uso di un Kernel ad hoc per questi sistemi. Questo tipo di soluzione apre però la possibilità di incorrere in altri bug che quindi costringono ad ulteriori passaggi di correzione. Alla fine di tutti questi processi di correzione siamo stati in grado di riportare il cluster a prestazioni accettabili e sopratutto in linea con quanto stabilito in fase di progetto. Punto chiave per poter risolvere il problema è stato l’utilizzo dell’ambiente chroot, ormai consolidato per il nostro sito, che, slegando l’ambiente in cui funziona il middleware da quello che gestisce la piattaforma hardware, ha reso possibile l’utilizzo di componenti personalizzate senza avere impatto sull’ambiente disponibile ai job grid.

Riferimenti bibliografici

[1] https://twiki.cern.ch/twiki/bin/view/FIOgroup/TsiBenchHEPSPEC

[2] http://www.spec.org/cpu2006/

[3] http://products.amd.com/pages/opteroncpuresult.aspx

[4] http://forums.amd.com/forum/messageview.cfm?catid=12&threadid=90112

[5] http://en.wikipedia.org/wiki/Translation_lookaside_buffer

[6] A. Ciampa et al., Alcune tecniche per GRID e dintorni, CCR-40/2010/P, (2010).

[7] https://savannah.cern.ch/bugs/?26178