Gigamax – cache consistency

Jan Horabik

22.03.2005

Plan prezentacji

- Co to jest Gigamax?
- Organizacja pamieci
- Konstrukcja modelu
- Efekty
- Przykladowy kod

Co to jest Gigamax?

- Wieloprocesor stworzony przez En-core Computer Corporation
- Potrzeba weryfiakcji – symulacje nie wystarczaja
- Weryfikacja zgodnosci cache'y i bra-ku blokady

Architektura

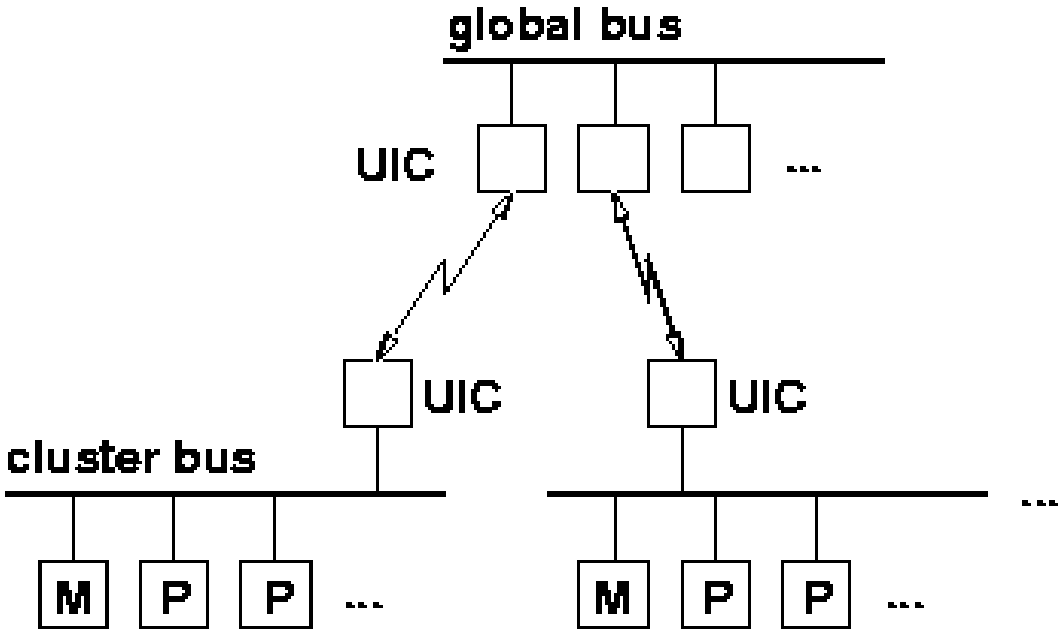

Figure 1: Gigamax memory architecture

Protokol

- Snooping cache
- ***“read miss”*** w cachu – wyslanie sygnalu read i oczekiwanie na odpowiedz
- Bus watcher – przechowuje informacje o adresach
- Stany bus watchera: Invalid, Shared, Owned, Xown (odnosnie kazdego adresu osobno)
- Reakcja – ***“memory bypass signal”*** - w przyszlosci ***“write response signal”*** (podwojne zadanie)
- ***“write invalidate”***

Zmiana wlasciciela

- Cache wysyla ***“read private”*** i zaklada blokade na swojego bus watcher'a
- bus watcher cache'a, ktory jest ***“Owned”*** przechwytuje sygnal, wysyla ***“memory bypass”*** i zmienia stan na ***“Xown”***
- Po jakims czasie wysyla ***“write response”*** i zmienia stan na ***“Invalid”***
- Pierwszy bus watcher otrzymuje sygnal, zdejmuje blokade, uaktualnia cache'a.

Bus watcher w UIC

- Posiada pelna informacje o lokalnych adresach i o tym, w ktorych cache'ach sa przechowywane kopie ich zawartosci
- Decyduje gdzie przesylac sygnaly
- Eliminuje koniecznosc wspoldzielenia jednej szyny przez wszystkie procesory

Stany bus watcher'a

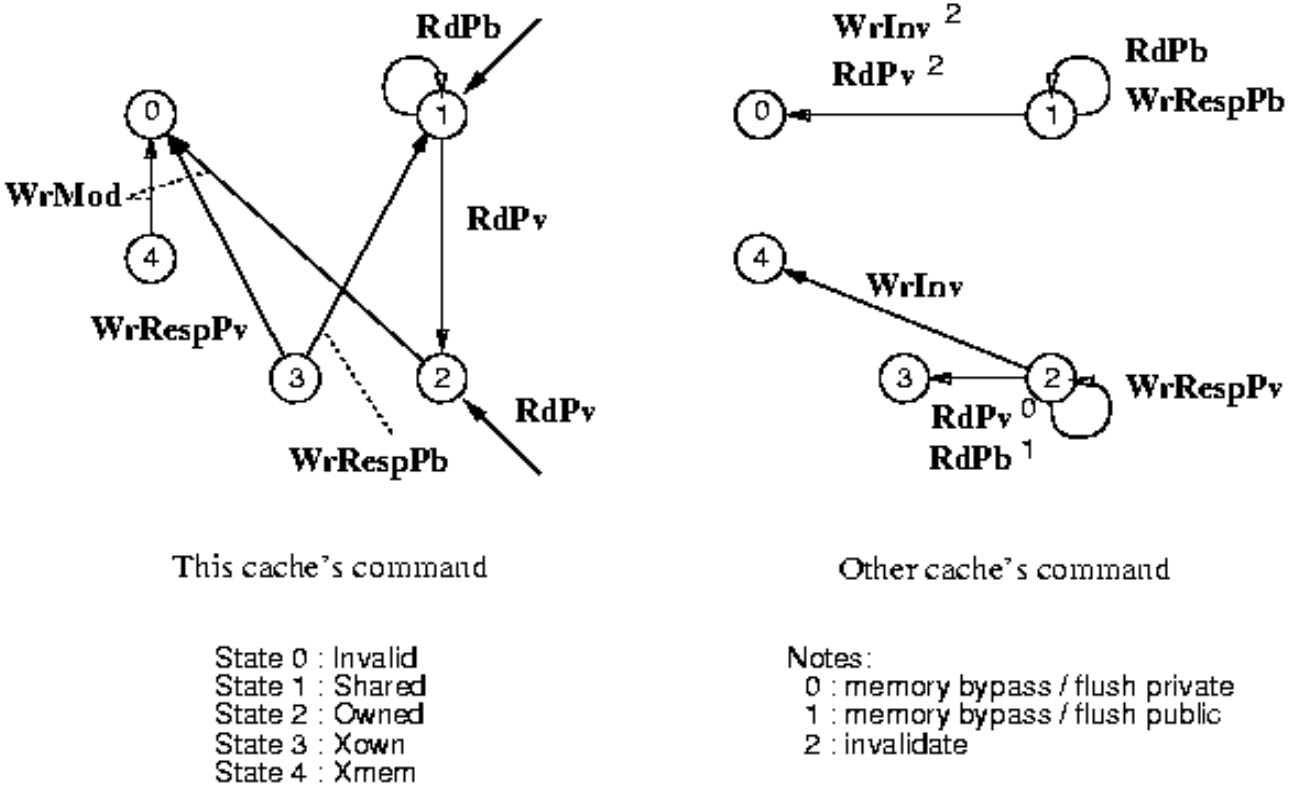

Figure 2: Bus watcher state diagram

Model

- Zajmujemy sie tylko jednym adresem
- Abstrakcja:

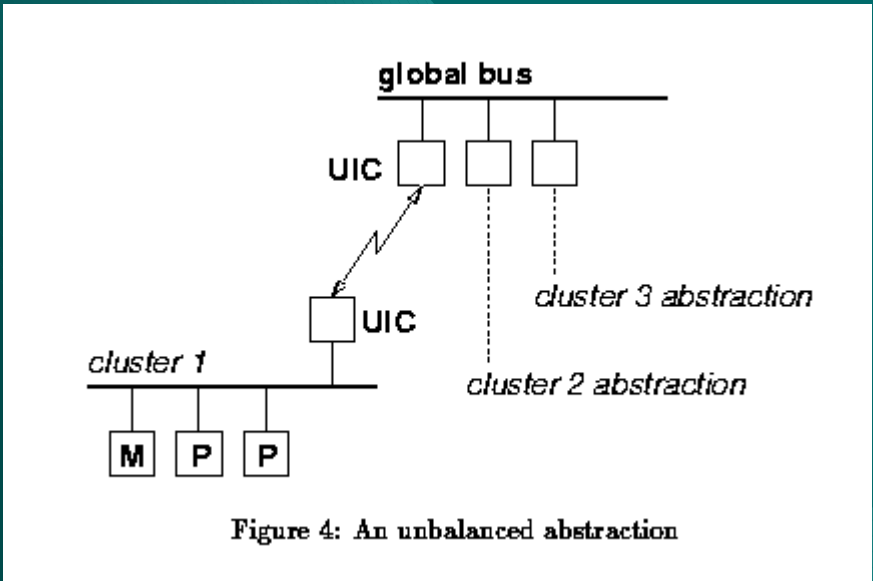

Figure 4: An unbalanced abstraction

P: stan bus watcher'a, 1 bit danych, stan blokady, lista komunikatow do cache'a.

UIC: lista komunikatow

link: 2 kolejki komunikatow

Co chcemy sprawdzic?

- Slaba zgodnosc cache'y: zapis do jednej kopii, odczyt z wielu kopii, uaktualnianych co jakis czas:

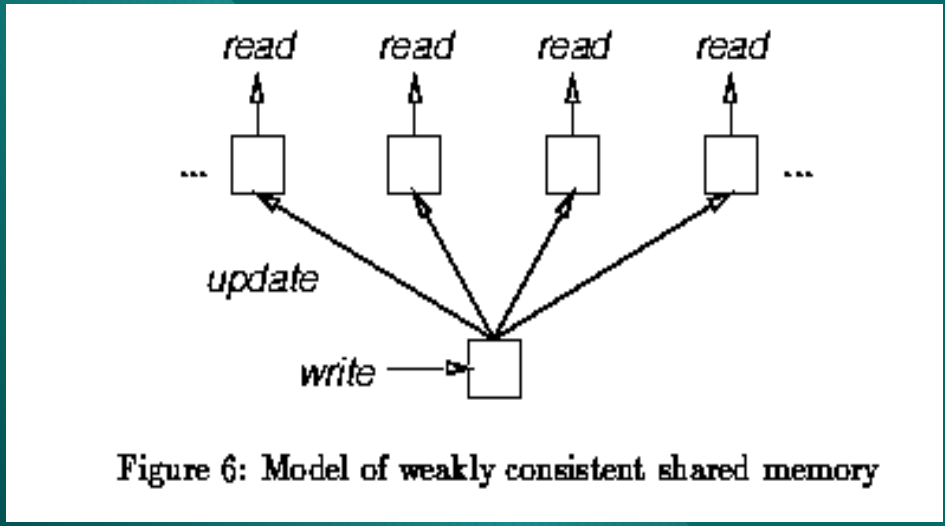

Figure 6: Model of weakly consistent shared memory

Co chcemy sprawdzi ?

- Brak sytuacjii blednych (bus watcher sygnalizuje b ad, gdy widzi nieoczekiwana sytuacje na szynie)
- Brak blokady (grupa bus watcherow nigdy nie osiagnie stanu ***“shared”*** lub ***“owned”***)
- Wszystkie zadania kiedys beda obsluzone

Wyniki

- Blokada:
  - Bus watcher procesora w klastrze 2 jest w stanie ***“owned”***, adres pochodzi z pamieci w klastrze 1.
  - Procesor w klastrze 1 wysyla ***“read public”***, ktore zostaje zakolejkowane. Bus watcher tego procesora zak ada blokade.
  - Procesor w klastrze 3 wysyla ***“read public”***, interweniuje bus watcher klastra 2 i wysyla ***“write response”*** do klastra 3. - oba bus watcher'y sa ***“shared”***, ***“write response”*** dociera do klastra 1 i pamiec zostaje zaktualizowana.
  - Procesor w klastrze 2 dokonuje zamiany cache'a – jego kopia jest juz nieaktualna
  - Procesor w klastrze 2 wysyla ***“read public”*** i zaklada blokade. Ten sygnal powinien zostac przekazany do klastra 1 (bo nikt nie jest w stanie ***“owned”***).
  - Pojawia sie blokada.