

GAZİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ



ÜMMÜ NUR GÜLMEZ

191180762

Araştırma Ödevi

PROF. DR. M. ALİ AKÇAYOL

Bilgisayar Mimarisi BM 311

Multicore ve Çok İşlemcili Sistemlerde Cache Coherence İçin Kullanılan Protokoller

İÇİNDEKİLER

1.ÖZET	3
2. CACHE COHERENCE.....	4
3.MULTICORE VE ÇOK İŞLEMCİLİ SİSTEMLERDE CACHE COHERENCE İÇİN KULLANILAN PROTOKOLLER.....	6
4.SONUÇ	8
5.KAYNAKÇA.....	9

1.ÖZET

Tek bir veri yoluna bağı çok sayıda çekirdek ve her çekirdeğin kendi içerisinde 1. Seviye önbellek bulunmaktadır, 2. seviye önbellek ise veri yolu üzerindedir. Bu sayede önbellek açısından her çekirdek kendi belleğini kullanıyorken veri yolu üzerinde paylaşılmış bir önbellek kullanımı sağlanmaktadır. Bu durum cache conherence'a yani önbellek tutarlılığının ortaya çıkmasına sebep olmuştur. Önbellek tutarlılığının sağlanması için donanımsal ve yazılımsal olarak iki çözüm yaklaşımıdır. Yazılımsal çözümlerde veriler işaretlenir ve önbelleklenemez duruma getirilir. Donanımsal çözümlerde ise protokoller geliştirilir. Snoopy ve Directory protokolleri donanımsal çözümler için geliştirilmiştir. Önbellek tutarlığı için geliştirilmiş olan protokoller MSI, MOSI, MESI ve MOESI olmak üzere dört adettir. Protokollerin baş harfleri ilgili önbellek bloğunun durumunu belirtmektedir. M harfi Modified, S harfi Shared, I harfi Invalid, E harfi Exclusive, O harfi ise Owned özelliklerini temsil etmektedir.

2. CACHE COHERENCE (ÖNBELLEK TUTARLILIĞI)

Çok işlemcili sistemler, birim zamanda bilgisayarın daha fazla iş yapması beklentisinden açığa çıkmıştır. Çok işlemcili sisteme sahip bilgisayarlar aynı ana bellek ve çevre birimlerini paylaşan iki veya daha fazla işlem birimi bulunduran CPU'lar için farklı cache belleklerin kullanılabildiği bilgisayar sistemleridir ve birden fazla işlemci aynı işin farklı kısımlarını gerçekleştirebilir. Çekirdekler işlemcileri yöneten ve bilgisayarın beyni olarak tanımlanan kısımlarıdır. Eş zamanlı veya farklı görevler üzerinde çalışabilen çekirdekler işlemcilerin merkezi yönetim sistemidir. Bir işlemcide çekirdek sayısı ne kadar fazlaysa işlemcinin çalışma kapasitesi de o kadar hızlı olur. Bilgisayarların çalışma kapasitesinin artırılmasının istenmesi sonucunda ortaya çok çekirdekli işlemciler çıkmıştır. [1][2][3]

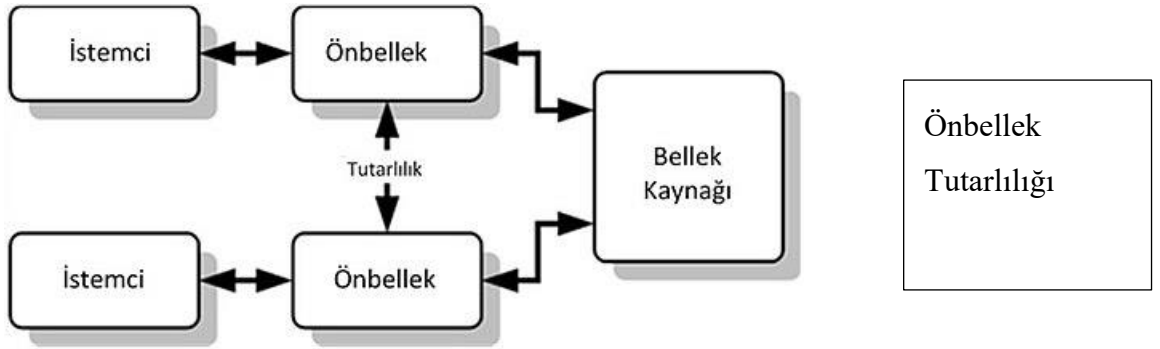


Tek bir veri yoluna bağlı çok sayıda çekirdek ve her çekirdeğin kendi içerisinde 1. Seviye önbellek bulunmaktadır, 2. seviye önbellek ise veri yolu üzerinde durmaktadır. Bu sayede önbellek açısından (cache), her çekirdek kendi belleğini kullanırken, aynı zamanda, veri yolu üzerinde paylaşılmış bir önbellek kullanımı sağlanır.

Şekil 1.1

Cache üzerinde bir verinin güncellenmesinden sonra ve verinin hafızaya ya da diske yazılmasından önce verinin işlem basamaklarının algoritma aracılığıyla izlenebilir - kaydedilebilir olması çoklu işlemcileri bulunan sistemlerde veri bütünlüğünü korumayı sağlamaktadır. Cache coherence, aynı verinin başka işlemciler tarafından aynı anda işlenmesini önlemek için mühendisler tarafından ortaya konmuş bir sistemdir. İşlemciler aynı verinin farklı önbelleklerdeki kopyalarını aynı anda

değiştirirse tutarsızlık ortaya çıkar. Veri üzerindeki değişikliklerin tutarlılığı bozmayacak şekilde yapılması gerekir. [4]



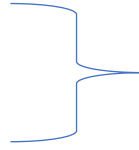
Şekil 1.2

Birden çok işlemci paralel olarak çalıştığında birden çok önbellek aynı bellek bloğunun kopyalayabilir ve bu durum cache coherence problemini ortaya çıkarır. Önbellek tutarlılığı sorununun çözümleri genellikle, satır değiştirildiğinde yinelenen satırların biri hariç tümünün geçersiz kılınmasıyla gerçekleştirilir. Gözetleme, önbellek tutarlılığı sorununu çözmede kullanılan mantıktır. İşlemciye gözetleme mantığı, önbelleğindeki bir sözcük her değiştirildiğinde veri yolu üzerinden bir mesaj yayımlar. Gözetleme, diğer işlemcilerden gelen bu tür mesajları arayan veri yolunu da gözetler. Bir işlemci, başka bir işlemcinin kendi önbelleğinde bulunan bir adresteki bir değeri değiştirdiğini algıladığında, gözetleme mantığı önbelleğindeki bu girişi geçersiz kılar. Bu duruma cross invalidate denir. Çapraz geçersiz kılma, işlemciye önbellekteki değerin geçerli olmadığını ve doğru değeri başka bir yerde (bellek veya diğer önbellek) araması gerektiğini hatırlatır. [5][6]

3. MULTICORE VE ÇOK İŞLEMCİLİ SİSTEMLERDE CACHE COHERENCE İÇİN KULLANILAN PROTOKOLLER

Önbellek tutarlılığını sağlayabilmek için iki tür çözüm vardır. Bunlar donanımsal ve yazılımsal çözümlerdir. Yazılımsal çözümlerde kod analiz edilir, önbellek için güvenli sayılmayan veriler işaretlenir. Donanım işaretlenen verileri önbelleğe almaz. Kısaca paylaşılan tüm veriler önbelleklenemez yapılmış olur. İkinci çözüm türü donanımsal çözümlerdir. Bu çözüm türünde cache coherence'ın sağlanabilmesi için protokol geliştirilir. Problem ortaya çıktığı an çözülür bu sebeple yazılımsal çözümlere göre daha iyi performans sağlamaktadır. [7]

- Directory Protokolleri
- Snoopy Protokolleri



Donanımsal Çözümler

Directory Protokolleri

Verilerin kopyasının nerede olduğunu öğrenir ve bellekte saklar. Herhangi bir önbellekte bulunmakta olan verinin üzerinde yapılması planlanan lokal işlem merkezi denetleyiciye bildirilir. Merkezi denetleyici verinin yer aldığı CPU'nun bilgisini tutar. Herhangi bir önbellek veriye yazma yapacağı zaman merkezi denetleyiciden izin ister. Diğer önbellekler geçersiz kılınır. Başka önbellek aynı veriyi istediğinde, merkezi denetleyici write back yapılmasını ister. Bu protokol türünde overhead oluşur. [7]

Snoopy Protokolleri

Her önbellek diğer önbelleklerle paylaştığı verileri bilir. Paylaşılan bir veride güncelleme yapıldığında tüm önbelleklerde broadcast bildirimi yapılır. Bu protokol türü bus trafiğini arttırır. [7]

Önbellek tutarlılığı için kullanılan protokoller şunlardır:

- MSI Protokolü
- MOSI Protokolü
- MESI Protokolü
- MOESI Protokolü

MSI Protokolü

MSI protokolünün açılımı Modified (Değiştirilmiş), Shared (Paylaşım) ve Invalid (Geçersiz)'dir. Bu açıklamalar önbelleğin olabileceği durumları tanımlamaktadır. Modified, önbellekteki verinin ana bellekteki veriden farklı olmasını temsil eder. Shared, verinin en az bir önbellekte bulunmuş olduğunu temsil eder. Invalid, veri geçersizdir ve geçersiz olduğu ön belleğin bağlı olduğu işlemcinin bu veriyi kullanabilmesi için, bu verinin ana bellekten veya bir başka ön bellekten getirilmesi gerektiğini temsil etmektedir.[9] [10]

MOSI Protokolü

Bu protokol, MSI protokolünün uzantısıdır. MSI'den farkı mevcut işlemcinin bu bloğa sahip olduğunu ve sahip olduğu blok için diğer işlemcilerden gelen isteklere hizmet vereceğini gösterir. [10]

MESI Protokolü

Bu protokol için Modified tutulan verinin ana bellektekinden farklı olduğunu belirtmektedir. Önbelleğin geçersiz ana bellek durumunun başka şekilde okunmasına izin vermeden önce verileri gelecekte ana belleğe geri yazması gerekmektedir. Exclusive verinin yalnızca geçerli olan önbellekte olduğunu ve ana bellektekiyle eşleştğini göstermektedir. Shared verinin diğer önbelleklerde de depolanabileceğini göstermektedir. Invalid, önbellek satırının geçersiz olduğunu belirtmektedir. [10]

MOESI Protokolü

Bu protokolde Modified verinin en son ve doğru halini tuttuğunu, ana bellekteki kopyanın yanlış olduğunu ve başka bir işlemcide verinin olmadığını gösterir. Owned,

yalnızca bir işlemcinin verilere sahip olabileceğini diğer işlemcilerin verileri paylaşılan durumda tutması gerektiğini göstermektedir. Exclusive, verilerin en son ve doğru kopyasını tutar. Ana bellek kopyası aynı zamanda verilerin en güncel, doğru kopyasıdır, başka hiçbir veri kopyası tutmaz olduğunu belirtmektedir. Shared durumundaki önbellek satırı, verilerin en son ve doğru kopyasını tutar. Sistemdeki diğer işlemciler de verilerin kopyalarını paylaşılan durumda tutabilir. Ana bellek kopyası, başka hiçbir işlemci tarafından Owned durumunda tutulmuyorsa, verilerin en güncel, doğru kopyası olduğu anlamına gelir. Invalid durumundaki bir önbellek satırı, geçerli bir veri kopyası tutmaz. Geçerli veri kopyaları, ana bellekte veya başka bir işlemci önbelleğinde olabilir.[10]

4.SONUÇ

Önbellek tutarlılığı gelişen çok işlemcili ve çok çekirdekli sistemlerle beraber açığa çıkan bir sorundur. İki çeşit yaklaşımla çözülmesi beklenmektedir: Donanımsal ve yazılımsal. Donanımsal çözümlerde problem ortaya çıkar çıkmaz çözüm sağlandığından yazılımsal çözümlere göre daha avantajlıdır. Önbellek tutarlılığının sağlanabilmesi için ortaya konan protokoller ilgili önbelleğin durumu baz alınarak belirtilmiş olan durumların baş harflerinden oluşmaktadır. Bu protokoller MSI, MOSI, MESI ve MOESI olmak üzere dört adettir.

5.KAYNAKÇA

1. <http://kursatcakal.azurewebsites.net/Makale/Detay/74#:~:text=K%C4%B1saca%20%C3%B6zetleyecek%20olursak%20%C3%A7ok%20i%C5%9Flemcili,i%C5%9Flemci%20ayn%C4%B1%20i%C5%9Fin%20farkl%C4%B1%20k%C4%B1s%C4%B1mlar%C4%B1n%C4%B1>
2. <https://www.longlinestore.com/islemci-cekirdek-sayisi-nedir>
3. <https://bilgisayarkavramlari.com/2010/10/05/cok-cekirdekli-islemciler-multi-core-processors/>
4. https://tr.wikipedia.org/wiki/Önbellek_tutarlılığı
5. <https://www.ibm.com/docs/en/aix/7.2?topic=architecture-cache-coherency>
6. <https://www.sciencedirect.com/topics/engineering/cache-coherence>
7. <https://w3.gazi.edu.tr/~akcayol/files/CAL11Multiprocessors.pdf>
8. <https://www.geeksforgeeks.org/cache-coherence/>
9. <https://tr.wikipedia.org/wiki/MSI>
10. <https://www.geeksforgeeks.org/cache-coherence-protocols-in-multiprocessor-system/>