

Paralel Güç

NE ZAMAN donanım firmaları daha güçlü ürünleri sunsalar, yazılımlar tüm bu olanakları emip bitirirler. Bir 486DX-33, MS-DOS altında bir kelime işlemcisi çalıştırmak için yeterli olabilir; ancak yeni yazılımlar, bu yıl sonunda piyasaya sürülecek olan Intel P-6 işlemcisinin tüm olanaklarını zorluyor olacaklar. Eldeki 486'lar, video anaçlığı ile bağlantı gibi multimedya fonksiyonlarını işlerken, yetersiz kalacaktır.

Bunun da ötesinde, kişisel bilgisayarlardaki hedeflerden biri gerçek hayatı taklit edebilmektir. TV, bilerek olmasa da, şu an için gerçek dünyanın elektronik dünyadaki anlatımının bir standartını oluşturmaktadır. Bilgisayarlar hakkında pek bilgisi olmayan biri, bir monitöre baktığında bir programcı veya kullanıcı kadar heyecanlanmayabilir. Uzman olanlar, ellerindeki sınırları bilirler; yeni kullanıcıların bunu anlamasını bekleyebilirler mi? 5 yıl içinde PC'ler, Jurassic Park'ı yaratan bilgisayarların gücüne sahip olacaklar; daha iyi renk işlemi, üç boyutlu analiz, real-time (gerçek hızda) animasyon, yüzlerce megabyte bellek ve bir çeşit paralel mimari.

Günlüğümüzde paralel işlemciliğin bu kadar önem kazanmasının bir sebebi, elimizdeki yarı iletken teknolojisinin sınırlarının bilinmesidir. Çipleri çok küçültmezsiniz, çünkü kuantum etkileri devreye girer. Aynı şekilde işlemcilerin saatlerini çok hızlandıramazsınız. Bu teknolojinin sınırlarına yaklaşıldığı herkes tarafından kabul edilmektedir. Ancak, önemli olan soru bir çipe kaç milyon transistör daha sıkıştırılabileceği değil, ekonomik olarak bugün neler yapılabileceğidir. Yani, paralel işleme günlüğümüzün teknolojisinin yerini alacak bir sistem değil; geleceğin gücüne uygun bir fiyatla sahip olmanın bir yoludur. 1 gigaFLOPS (1 giga floating point operations per second: saniyede bir milyar ondalıklı sayı işlemi) hızı ulaşmak için, ya 3 yıl daha beklemeyiz ya da günlüğümüzün RISC teknolojisine sahip bir kaç çipini paralel kullanmamız gerekecektir.

Paralel işleme, kısaca, bir büyük problemi küçük birimler kullanarak çöz-

mek olarak tanımlanabilir. Bu sayede, tek işlemcili bilgisayarlardaki gibi bilgileri bir-biri ardına işlemekten (seri işleme), birkaçını aynı anda işlemek mümkün olacaktır. Aslında bu tip bir paralel işleme; çiplerin içine de yavaş yavaş girmektedir. Yeni üretilen işlemcilerin çoğu bir saat döngüsünde birçok işlemi yapabilme kapasitesine sahip. Süpercalar teknolojisi ile üretilen çiplerin (Bkz. Bilim ve Teknik Şubat'95 Bilgisayar Dünyası) mikroparalelizm olarak da adlandırılan tasarımları da buna benzer bir özellik göstermekte. Paralel işleme, şu an kullandığımız PC'lerde bile sınırlı olarak mevcut aslında. Görüntü, I/O ve ses kartlarındaki çeşitli işlemciler, ana işlemci ile paralel olarak çalışmaktadır. Ancak, bu özel amaçlı bir paralelliktir ve matematik işlemlerinin daha hızlı yapılmasına pek bir katkısı olmaz.

Genel amaçlı paralel işleme, iki ana gruba ayrılabilir: SIMD (Single Instruction / Multiple Data: Tek Komut/Çok Bilgi) ve MIMD (Multiple Instruction / Multiple Data).

SIMD bilgisayarları, bir işlemi birçok bilgi üzerinde, aynı anda yapabilmek üzere tasarlanmıştır. SIMD'nin kullanımı için en basit örnek, bir matris üzerinde yapılacak işlemler olacaktır. Burada, matrisin tüm elemanlarına aynı işlem uygulanacağından, paralel işleme mümkün olacaktır. SIMD sistemleri, genel amaçlı bilgi işlemeye en iyi çözüm olmasalar da, dar kapsamlı bir paralellik için ucuz ve yeterli sistemlerdir. Bu sistemler, genellikle, bir



GigaCache modeldeki, 1A,084 işlemci işlemci devreleri oluşturmaya üzere üretilen bir dizi devredir.

kontrol merkezi ile ona bağlı basit işlemler yapabilen işlemcilerden oluşmaktadır. SIMD sistemleri, sadece büyük ve güçlü bilgisayarlarda değil, masa üstü bilgisayarlarda şekil tanıma ve bilgi sıkıştırma gibi özel amaçlarla da kullanılmaktadır.

SIMD bilgisayarları, mühendislerin sonlu eleman analizlerinde, jeologların sismik modellemelerinde, meteorologların atmosfer modellemelerinde ve daha birçok farklı amaçta kullanılan matris işlemlerini hızlandırmak için yeterlidir. Ancak, birçok farklı işlemlerin paralel olarak yapılmasını gerektirecek genel hesaplamalar için uygun değildirler.

SIMD'nin alternatifi olan MIMD ise, farklı programların veya bir programın farklı bölümlerinin çalıştığı birçok işlemcisi birbirlerine bağlamaktan ibarettir. Bu durumda, işlemciler arasındaki iletişim öne çıkmaktadır. Eğer iki işlemcinin bir problemin farklı yerlerini işlerken yardımlaşmaları gerekirse iletişim kurulmalıdır. Bu iletişimin nasıl olduğu gözünüze alınarak, MIMD bilgisayarları da ortak bellekli ve ayrı bellekli olmak üzere gruplandırılabilirler.

Ortak bellekli çokişlemcilerde (MIMD'lerde) tüm işlemciler ortak bir bellek deposuna ulaşırlar. Bu ulaşım da yine ortak, çok hızlı bir data bus anaçlığı ile olur. Teoride, bu işlemciler arasındaki iletişimi kolaylaştıracağı; her işlemci kendi bulunduğu sonucu belleğe koyar ve diğerlerine bilgiyi hangi adreste bulacaklarını söyler. Ancak teori ne kadar iyi olursa ol-

sun, uygulama o kadar kolay gözüküyor. Çok hızlı birçok CPU, ortak belleğe ulaşımında, data bus hızının sınırlamaları nedeniyle kaplumbağa ile yarışacak duruma gelebilir. CPU'ların sahip oldukları cache'ler (iç bellekler), onlara, bilgiyi sunmaları gerekene kadar kendi içlerinde tutma şansını sağlar. Bu durumda da aynı değişkenin farklı değerleri CPU'ların cache'lerinde ve ana bellekte yer alabilir. Bu engeli aşmak için de data bus'ün sürekli kontrolünü içeren bir sistem geliştirilmiştir.

Bu sorunlardan dolayı ortak bellekli bir MIMD'de yer alabilecek işlemci sayısı sınırlıdır. Yüzlere belki ulaşabilir, ancak süper bilgisayarların ihtiyaç duyduğu binlere hiçbir zaman çıkamaz. Tipik ortak bellekli MIMD'de, bir UNIX işletim sistemi ile her işlemcide bir UNIX işinin koştuğu kaba bir paralellik kullanılır. Bu programlamayı kolaylaştırır; işletim sistemi işleri CPU'lara dağıtacağı için kullanıcı hızlı bir seri UNIX makinesi kullandığını sanacaktır. DOS tipi işletim sistemleri, multiprocessing yapamamaları nedeniyle bu makinelerde kullanılmayacaktır.

Ayrı bellekli MIMD sistemlerinde ise, her hesap noktası kendi belleği ile tam bir bilgisayardır. Bu makinelerde multicomputer de denilmektedir. Hiç bir nokta diğerinin bellek alanına müdahale edemeyeceğinden, sonuçlar bir iletişim ağı içinde noktalar arasında aktarılmalıdır.

Bir multicomputer'in performansı işlem yapan elemanlara olduğu kadar iletişim ağının hızına da bağlıdır. Daha doğrusu, hesaplama zamanıyla iletişim zamanı arasındaki denge, kullanılan algoritmaya ve ağına topolojisine ve performansına bağlı olarak, problemde probleme değişecektir. Makine iletişimde zaman harcarken işlemcilerin yarısını beklemesi gerekebileceğinden, bir probleme çok işlemci atamak, o problemi gözünce en hızlı yolu olmayabilir. Bu nedenle uygun bir programlamanın yanında iyi bir iletişim ağı oluşturulması gerekecektir. Günlüğümüzde birçok ağ şekli denenmiştir; hatlar, diziler, ağaçlar, yürekler, 4 boyutlu hiperküpler, 8 boyutlu hiperküpler ve dahaları. Herbir problem için farklı bir ağ şekli en uygunu olabilir.

NEDEN PARALEL?

Sorun: 1 gigaFLOP'luk bilgisayar gücüne ihtiyacımız var.
Çözümler:

Üç yıl bekleyin.



Bugün elinizdeki.



Ancak, sabit bir topoloji ile tüm işlemcileri birbirine bağlamakta, onları iletişim için uygun bir ağ ve hızlı yönlendirme ile sağlamak daha ekonomik ve hızlı olmaktadır.

Multicomputer'lerde, hızları nedeniyle RISC işlemcileri kullanılmaktadır. Ancak ulaşılabildikleri yüksek hızların tek nedeni bu değildir. Ortak bellekli MIMD'lerin belirli bir işlemci sayısından sonra hızlarının pek artmadığına değinilmiştir; oysa, bir multicomputer'ün hızı, ona konulan çip sayısıyla yaklaşık olarak aynı oranda artacaktır.

Ticari alanda bu tip makinelerin yaygın olarak kullanılmasını engelleyen en büyük etmen, programlanmalarının oldukça zor olmasıdır. Bu nedenle multicomputer yazılım piyasası kurulamamıştır. Bu makineler sadece araştırma merkezleri ile üniversitelerde kullanılmaktadır. Multicomputer konusunda araştırmalar yapan şirketler ise, onları pazara sokabilmek için yollar aramaktadırlar. Birincisi, daha iyi yazılımlar sunmaktır. İkincisi ise, bir sunucu-müşteri ortamında kullanılmaktır. Bu sayede sunucu multicomputer bir kara kutu olarak kalacaktır. Kullanıcıların onun nasıl çalıştığını bilmelerine hiç gerek olmayacaktır.

Multicomputer'lar günümüzün en güçlü bilgisayarları durumundadırlar. Bir kaç yüz ile bir kaç bin CPU, tek bir kasada, yüzlerce gigabyte (1 gigabyte=1000 megabyte) bellek ile birlikte. Bunların her birinin inandırılmaz hesaplama gücü ile en zor problemler çözülebilmektedir. Bu noktada değinilmesi gereken konu ise, onların maliyetidir. Bir büyük multicomputer için 10 milyon dolardan fazla paranın gözden çıkarılması gerekecektir. Ülkemizdeki üniversitelerin bir ikisi dışındakiler, araştırmaları için bu miktarın bir kısmını bile sahip olamamaktadır. Örneğin, nükleer enerji mühendisliği konusunda ciddi araştırmalar yapılabilmesi için gereken büyük miktardaki hesaplama gücünün bu durumda karşılanmasına olanak yoktur. Eldeki olanakları hiç olmazsa günlük işlemlerin daha fazlasını sunabilecek hale getirmek için bazı yöntemlere başvurulması gerekmektedir.

Dağıtılmış hesaplama yöntemi, bu noktada güçlü paralel makinelere ekonomik bir alternatif olarak karşımıza çıkar. Dağıtılmış hesaplama, bir network aracılığı ile birbirlerine bağlı olan birçok bilgisayar, büyük bir problemi çözmek üzere birarada kullanılmaktadır. Bu sistem, multicomputer'lerin bir modeli gibidir. Aynı şekilde hesap noktaları arasında mesajlar aktarılabilir. Bu sayede bir networkün sanal bir paralel makine gibi kullanılması olasıdır.

Bir multicomputerde her bir işlemcinin kapasitesi, yazılım ve iletişim hızı aynıdır. Oysa, dağıtılmış hesaplama yapılan bir yerde PC'ler, iş istasyonları ve daha büyük bilgisayarlar bulunabilir. Bu durumda, yapı, bilgi formatı, hesaplama hızı, makinelerin yükü ve network yükü göz önüne alınması gereken farklılıklardır. Bu nedenle bu sistemlere heterojen network hesaplaması da denilmektedir.

Heterojen network hesaplamalarının bazı avantajları vardır. Eldeki donanımın kullanılması nedeniyle hesaplama için harcanacak para düşük olacaktır. Performans, her bir görevin ona uygun makinede çalıştırılması ile artırılabilir. Heterojen yapıyla dolayısıyla gerekli noktalarda lokal olarak farklı bilgilerin kullanılması mümkün olmaktadır. Sanal bilgisayarın parçaları değiştirilebilir; böylece makine bilgisayar teknolojisindeki gelişmeleri izleyerek geliştirilebilir. Sanal bilgisayar büyütülebilir. Her makinenin kendi ortamı kullanılır. Her bilgisayar üzerinde yeterli bilgiye sahip kişiler vardır. Kullanıcı ve program hataları yerel olacağından kolayca düzeltilir. Ve ortak çalışma yapılabilir.

Bu sistemin dezavantajları da yok değil. Bir makinenin üzerindeki kullanıcı yükü arttığında, o makine yavaşlayacak ve heterojen network hesaplamasına yeterli miktarda güç aktaramayacaktır. Bu durumda tüm program bir süre için beklemek zorunda kalabilecektir. Aynı şekilde, her bir makineye ne kadar yüklenileceği de, ayrı bir dikkat gerektirecektir.

Heterojen network hesaplamaları için günümüze kadar yazılımlar geliştirilmiştir. Argonne National Laboratory (ANL) tarafından geliştirilen p4 sistemi,

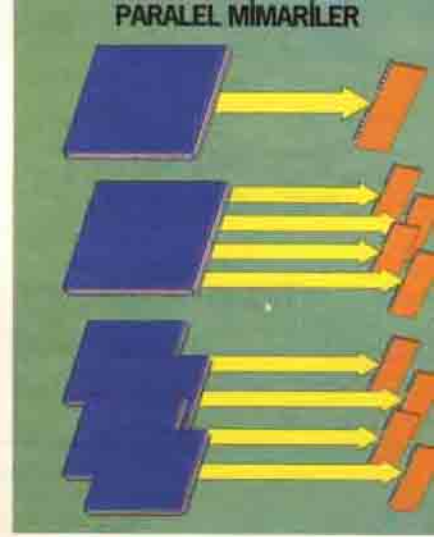
hem ortak bellekli bir makine gibi, hem de bir multicomputer gibi davranabilir. Bu sistemde bir yönetici iş, köle işleri kontrol ediyor. Express adlı bir başka sistem ise, ticari olması nedeniyle kullanıcılarına birçok olanaklar sunuyor. Sıralı bir programın nasıl uygun bir şekilde paralelleştirilebileceğini de inceleyen program, daha sonra bunu uygulayarak paralel programı elde ediyor. Linda sistemi ise, yaratıcı bir sanal sistemde iletişimi sağlıyor. Kullandığı yöntem, eldeki bellekleri bir noktaya kadar ortak olarak kullanmak ve ortak bilgilerin değiştirilmesini mümkün kılacak gibi olanaklar sağlıyor.

Bir başka dağıtılmış hesaplama yöntemi olan PVM (Paralel Virtual Machine: Paralel Sanal Makine), ABD'de yürütülen Heterojen Ağ Projesi çerçevesinde hazırlanmıştır. Bu, çok farklı yapıları ve öndelikli sayı tanımlara sahip bilgisayarların bir arada kullanılmasını sağlayan ilk yazılım sistemlerinden biridir. Küçük ve kolayca sisteme yerleştirilebilir olması da bu sistemin çok tercih edilmesinin bir nedenidir.

PVM'in kullanıcı arabirimi tüm mesaj bilgilerinin ayrıntısı ile belirtilmesine ihtiyaç duymaktadır. Gerektiği durumlarda makine tipinden bağımsız olarak bilgi formatını değiştirerek farklı tiplerdeki makineler arasında iletişimi sağlayabilmektedir.

PVM'in hesaplama modeli, bir uygulamanın çeşitli birimleri olduğu düşüncesini temel almaktadır. Her birim, uygulamanın hesaplama işlerinin bir kısmından sorumludur. Bazen, bir uygulama paralel birimlere ayrılabilir; yani her birim farklı bir görevi yerine getirir; girdi, problemin hazırlanması, çözüm, çıktı ve sunuş gibi. Bu işleme görsel paralellik de denilmektedir. Parallellığın daha çok kullanılan bir yöntemi de bilgi paralelligidir. Bu metoda, tüm birimler aynıdır; ancak her birim programın bir kısmını bilir ve çözer. PVM her iki sistemi de desteklemektedir. Görevlerine bağlı olarak, birimler paralel olarak çalışabilir, senkronize olabilir veya bilgi iletişimi içinde bulunabilir.

PVM'in bir özelliği de kullanıcıyı kolaylıktır. Bir kullanıcının kendi PVM'ini oluşturan donanım bir networke



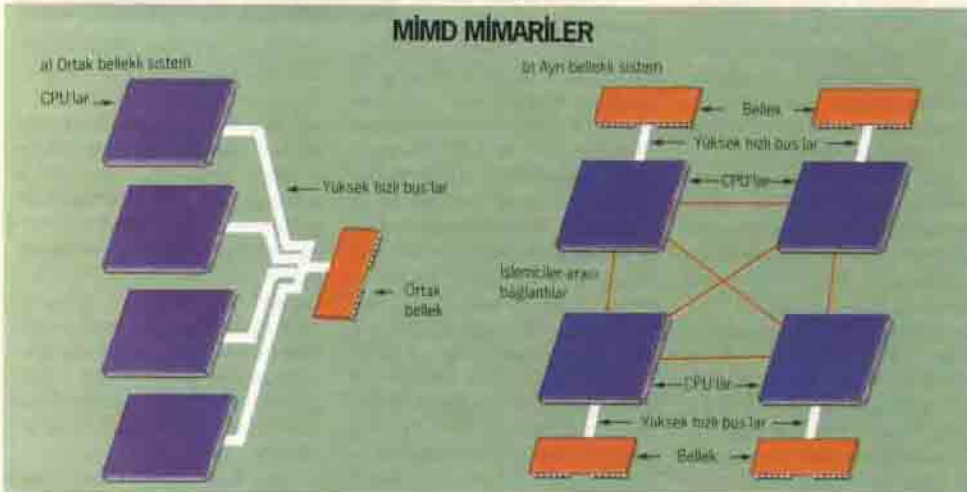
a) Seri bilgisayar sisteminde, bir komut, bir veri parçası üzerinde çalışır. b) SIMD makinelerde, birçok veri parçası, bir komut ile ya da farklı işlemcilerdeki aş komutlarla, aynı anda işlemi. c) MIMD makinelerde, bir çok işlemci, farklı veri parçaları üzerinde birbirinden bağımsız olarak işlev görürler.

bağlı, herhangi bir UNIX makine olabilir. Her kullanıcı PVM'i kullanarak kendi sanal paralel makinesini tanımlayabilir. Bir paralel sanal makineyi hazırlamak için PVM çalıştırıldıktan sonra istenilen makinelerin isimlerini girmek yeterli olacaktır. Fortran veya C kullanılarak yazılmış uygulamalar, multicomputerlerdeki gibi mesaj aktarım yapılarının eklenmesi ile paralelleştirilebilir. Bu mesajların aktarımı ile, bir uygulamanın farklı birimleri, bir problemi paralel olarak çözmek için iş birliği yapabilirler.

Bu sistemin daha rahat çalıştırılmasını sağlayan bazı ara birimler de zaman içinde eklenmiştir. Birincisi, X Windows altında çalıştırılabilen, grafik temelli xpm'dir. Bu programla, kullanıcının istediği PVM sistemini; programları hazırlaması ve daha sonra bunları kontrol etmesi oldukça kolaylaşmaktadır. Bir başka ara program olan HeNCE (Heterogeneous Network Computing Environment: heterojen network hesaplaması ortamı), heterojen bir ağ üzerindeki programların yazılmasını, derlenmesini, koşulmasını ve hatalarının düzeltilmesini kolaylaştırmak üzere hazırlanmıştır. Hedefleri, paralel hesaplama konusunda yeterli eğitimi olmayan bilim adamlarının ve mühendislerin network hesaplaması yapabilmelerini sağlamak ve onların hesaplamaların ihtiyacına uygun kaynakları kullanabilmelerini sağlamaktır. xpm gibi X Windows altında çalışan HeNCE kullanıcısının yapması gereken, bir programı grafikler aracılığı ile tanımlamaktır.

Yurt dışında birçok üniversite ve araştırma kurumu, bu sistemleri, ellerindeki kaynakları en verimli bir şekilde kullanabilmek için yerleştirmektedir; oysa ülkemizde bu durum biraz daha farklıdır. Bizler, elimizdeki donanımların bir miktar yeterli olabilmesi için bunları kullanmak durumundayız.

Kaynaklar:
G.A. Geist, A. Beguelin, J.J. Dongarra, W. Jiang, R. Manček ve V.S. Sunderam, PVM: Parallel Virtual Machine, The MIT Press, Cambridge, Ma (1994).
BYTE, Ağustos, s:112-136 1992.
J.J. Dongarra, G.A. Geist, R. Manček ve V.S. Sunderam, Integrated PVM Framework Supports Heterogeneous Network Computing, 3 Ocak 1993.
G.A. Geist, V.S. Sunderam, Network Based Concurrent Computing on the PVM System.



a) Ortak bellekli sistemlerde, çok işlemciler, ortak bir bus üzerinden, ortak bellek deposundan geçerek iletişim kurarlar. b) Ayrı bellekli sistemlerde, her işlemcinin kendi bellek deposu vardır ve yüksek-hızlı bağlantılar aracılığıyla iletişim kurar.