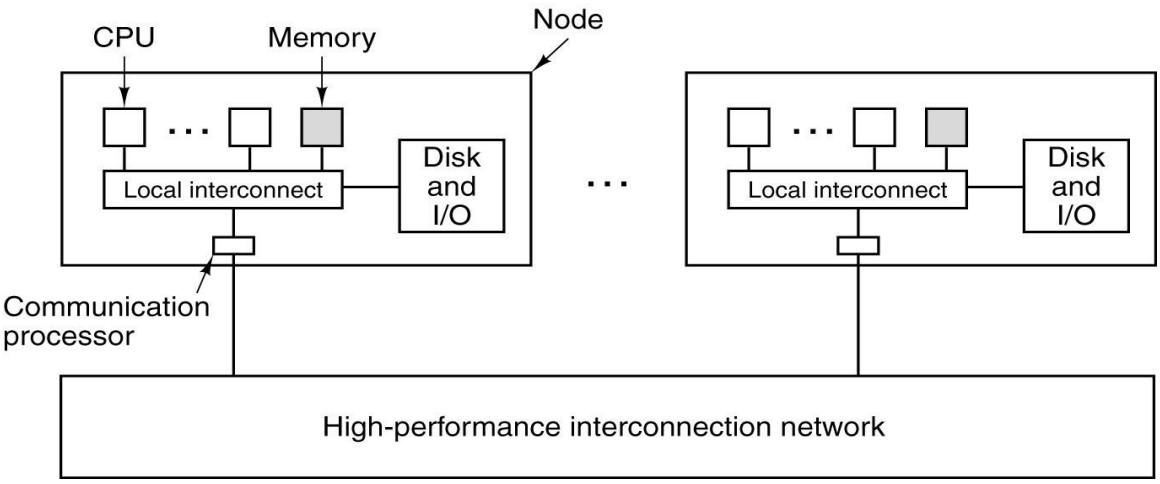
8.4) Message Passing Multicomputers (Mesajlarla anlaşan çoklu bilgisayarlar)

Bir işlemi birden fazla iş parçacığına bölerek birbirine az bağımlı ya da bağımsız iş parçacıklarını birden fazla bilgisayara dağıtıp işlemi eşzamanlı olarak farklı bilgisayarlarda yapan ve bağımlılıkları da bilgisayarlar arasındaki yüksek hızlı dahili ağda mesajlaşarak halleden bu sisteme mesajlarla anlaşan çoklu bilgisayar sistemi denir.



Bu sistemler genellikle efendi (master) ve köle (slace) ilişkisi içinde gerçekleşir. Köle işleyeceği kodu efendisinden alıp işler. Gerektiğinde de işleyeceği bilgiyi yine efendisinden ister. Efendisi de istenen bilgiyi gönderir ve gönderdiği bilgiyi de işaretler. Bilgi işlenip geri gelene kadar efendi ve diğer köleler tarafından kullanılmaz.

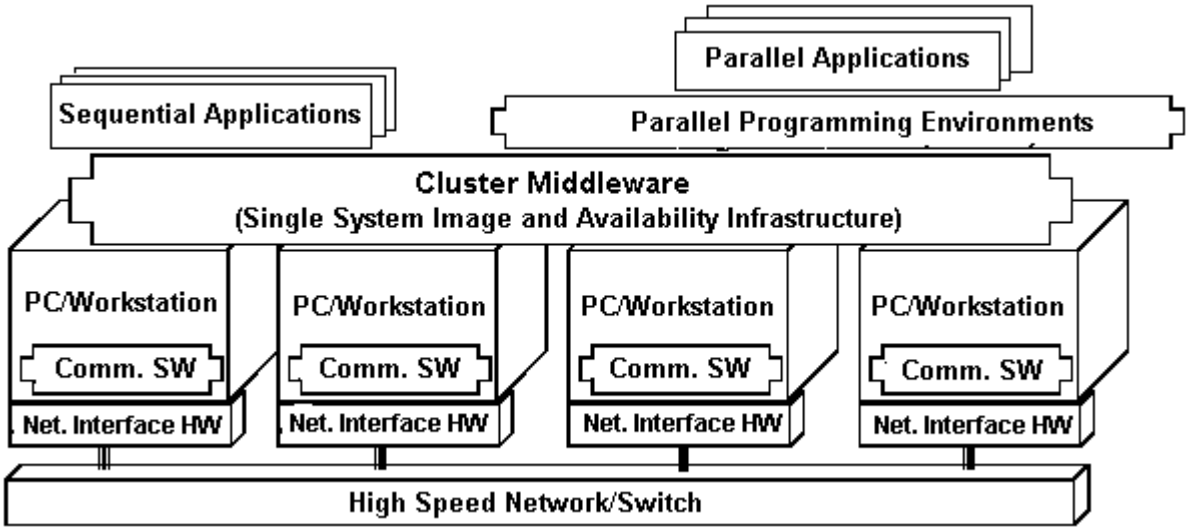
Bu arada çoklu işlemcili sistemlerle çoklu bilgisayarlı sistemleri karşılaştırmak gerekirse, çoklu işlemcili sistemlerin yapımı (build) daha zor; ama programlanması kolayken, çoklu bilgisayarlı sistemlerin de yapımı çok kolay; fakat programlanması çok zordur. Önemsenmesi gereken bir konu da çoklu işlemcili sistemlerde farklı işlemler (ya da işlemcikler [threads]) aynı veriye ulaşmak zorunda kaldıklarında, bir başka deyişle bağımlılıkları arttıklarında performans kayıpları üst düzeye çıkar. Çoklu işlemcili sistemlerde yükle (load) ve sakla (store) komutları kullanılırken, çoklu bilgisayarlı sistemlerde bu komutlar yerine gönder (transmit) ve al (receive) komutları kullanılır. Çoklu işlemcili sistemler ile çoklu bilgisayarlı sistemler arasında belli başlı farklar olsa da benzer temel üzerine inşa edildikleri de söylenebilir, çünkü iki sistem

de işlemler arası mesajlaşma prensibine dayanır. Biri aynı bilgisayar üzerinde genellikle portlar üzerinden (belli bir bellek bölgesi ya da I/O cihazı da olabilir) mesajlaşırken diğer sistemde mesajlaşma ağ arayüzü ile yapılır.

8.4.3) Cluster Computing (Küme Bilgisayarlar)

Küme bilgisayarlar tipik olarak yüzlerce, binlerce kişisel veya iş istasyonu (Workstation) bilgisayarların birbirlerine ucuz maliyetli ağ alt yapısı ile bağlanmasından oluşurlar. MPP (Massively Parallel Processing) ile küme bilgisayarların arasındaki fark mainframe ile kişisel bilgisayar arasındaki farka benzer. MPP ile küme bilgisayarlarının ikisi de kendilerine ait işlemciye, belleğe, diske ve işletim sistemine sahiptirler. Ama mainframe'ler sadece daha hızlıdırlar.
Tarihsel olarak MPP'leri özel yapan anahtar element aralarındaki dâhili yüksek hızlı bağlantıdır. MPP'ler yüksek bütçeli süper bilgisayarlardır.

Küme bilgisayarların çeşitleri fazla olmasına rağmen merkezi ve merkezi olamayan şeklinde gruplanmışlardır. Merkezi bir küme kişisel bilgisayarların ve iş istasyonlarının bir odada bir rafta konumlanmasından oluşur. Tipik olarak bilgisayarlar bulundukları yere homojen olarak konurlar ve ağ kartlarından ve disklerinden başka çevresel birim bulundurmazlar. Merkezi olmayan kümeler ise bir binaya veya bir kampüse yayılmış heterojen vaziyettelerdir. Bilgisayarların çoğunun gün içinde iş yapmadıkları zamanları olur. Boş bir iş istasyonunu bir kümeyi yönetmek için kullanmak bu kümenin efendisinin işlerini kölelerine dağıtmasına olanak sağlar. Bu da yazılım karmaşıklığını arttırır.



Yukarıda Tipik Küme Biligisayar Mimarisi Görülmektedir.

Küme Bilgisayar Uygulama Alanları

- Coğrağfi Bilgi Sistemleri
- Görüntü İşleme
    - Optik Karakter Tanıma
    - Parmak İzi Tanıma
    - Özel Amaçlı Görüntü İşleme
- Fabrika Üretim Hattı Hata Denetimi
- Furye Dönüşümü ve Büyük Ölçekli Matris Hesaplamaları

Küme Bilgisayar Avantaj ve Dezavantajları

Avantajlar

1. Donanımını temin etmek kolay ve ucuzdur.
2. Tek bir donanım üreticisine bağlı kalma zorunluluğu yoktur.
3. Linux sürücüleri donanımları destekleme sorunu yaşamazlar.
4. Çok sayıda bilgisayar ile tek bir süper bilgisayarın hızına erişilebilir.
5. Küme bilgisayar mimarisi hatalara (çökmelere) karşı etkin bir koruma sunmaktadır.
6. Küme bilgisayar sistemine bir düğüm bilgisayarı ilave edilmek istendiği zaman bu bilgisayarın işletim sistemi ve yazılımını üstüne yükleyip sisteme ağ üzerinden entegre edilebilir. Bu iş çok kolaydır.

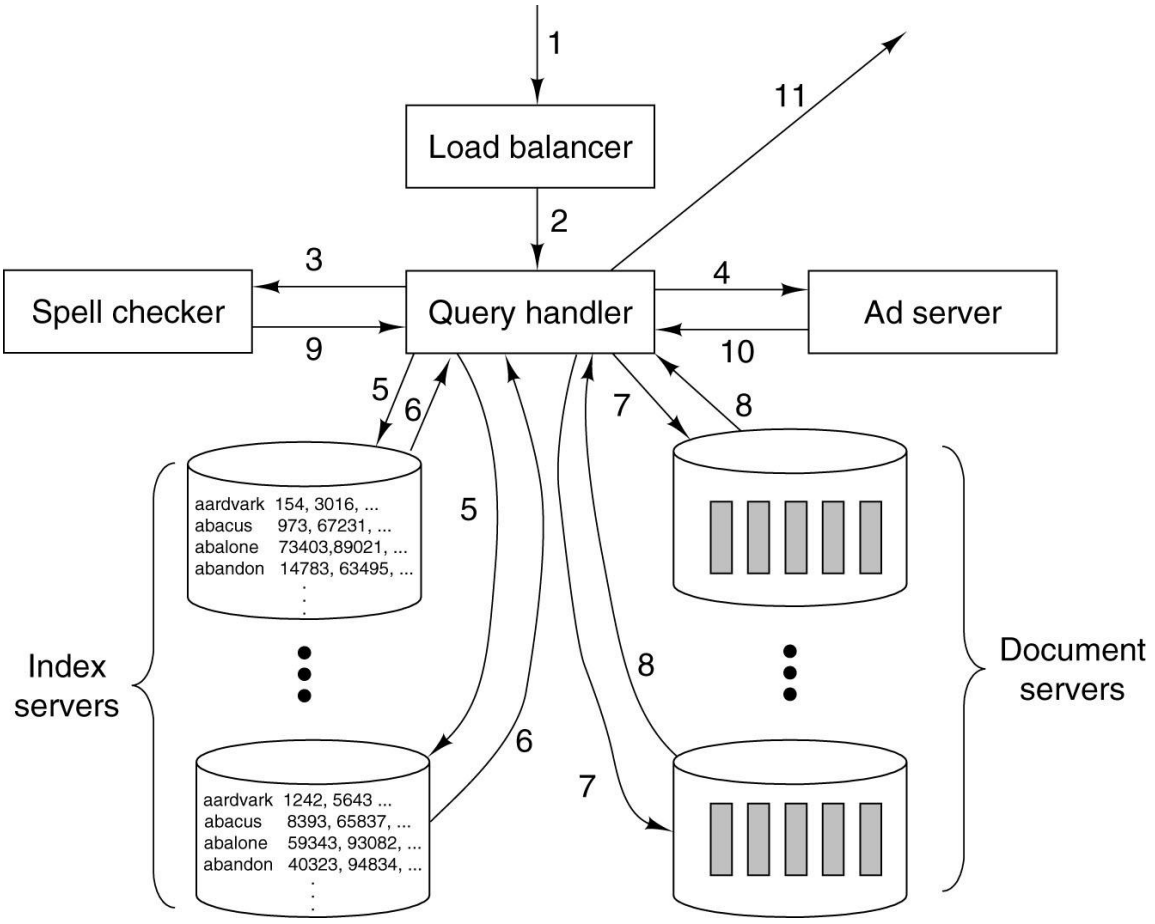
Dezavantajlar

1. Küme hesaplamaya verilecek işler doğaları gereği paralel işleme uygun olmalılar. Bunu yazılımsal olarak sağlamanın da bir maliyeti vardır.
2. Kullanılan kişisel bilgisayarlar donanımsal veya yazılımsal olarak çok kolay arıza yapmaktadır.

Gooogle'ın kümelemeyi kullanışı

Google açısından problem 8 milyar web sayfasını ve 1 milyar resmi bulmak, indekslemek, kaydını tutmaktır. Bunların yanında 0.5 saniyede birçok sorguyu gerçekleştirmek ve bu sistemi de 7/24 çalışmasını sağlamaktır. Ayrıca bu sistem depremlerden, elektrik kesintilerinden, donanımsal problemlerden ve yazılımsal sorunlardan etkilenmemelidir. Bunu yaparken de mümkün olduğunca maliyeti düşürmelidir.

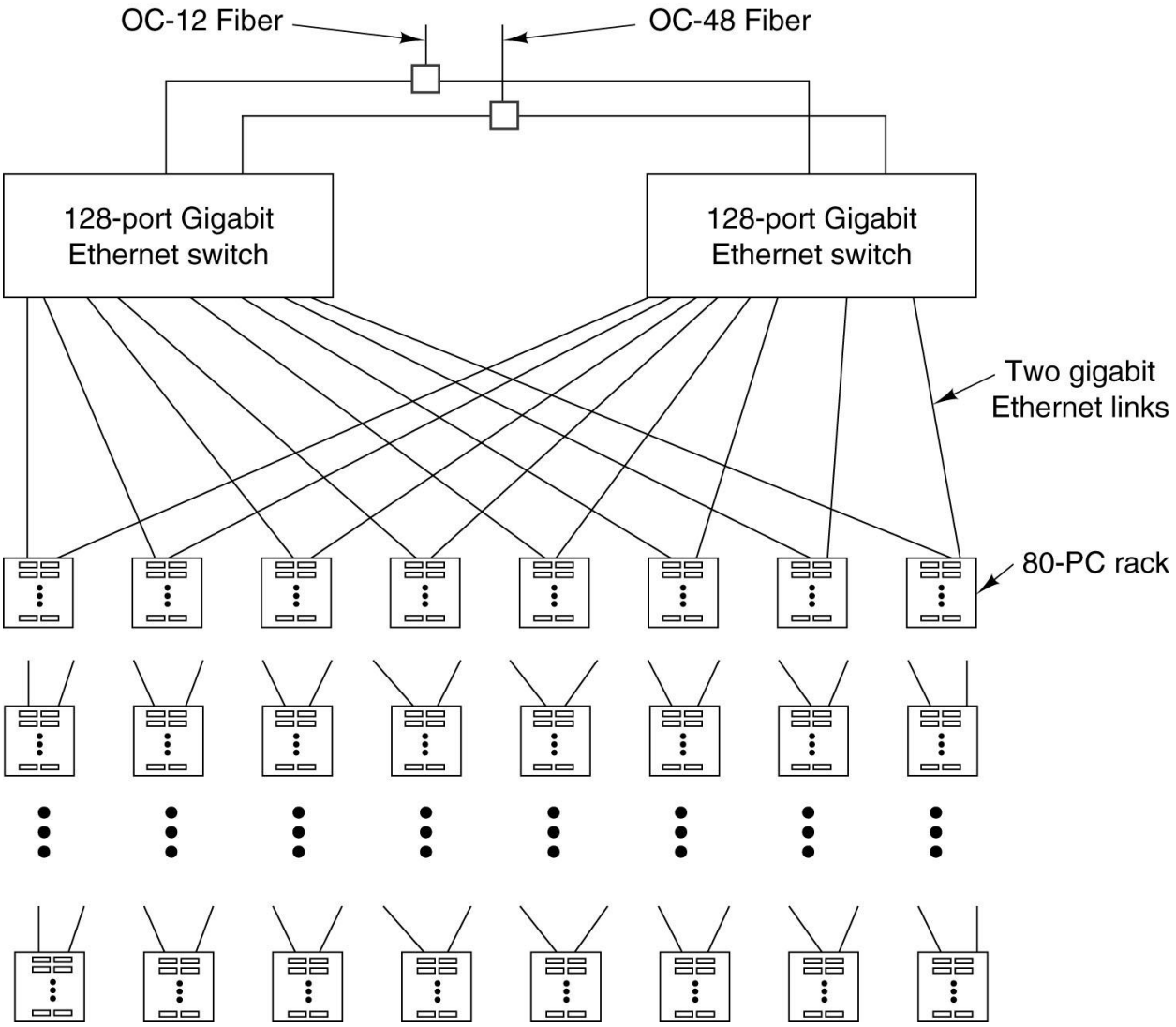
Başta Google dünyanın çeşitli yerlerinde çoklu datacenter'a sahiptir. Her bir datacenter en azından OC48 (2.488 Gbps) fiberoptik internet bağlantı hızına ve OC12 (622 Mbps) yedekleme bağlantı hızına sahiptir. Yedekleme yapılan datacenter da başka bir internet servis sağlayıcı ile internete bağlıdır.



Bir sorgunun işlenişinin tarif edilmesiyle Google'ın bu mimariyi seçmesinin sebebi daha iyi anlaşılabilir. Sorgunun datacenter'a ulaşmasıyla yükleme dengeleyicisi (load balancer) alınan sorguyu sorgu işleyicilerinden (tutucularından) (query handler) uygun (boş) olanına yönlendirir(2). Sorgu işleyicisi heceleme kontrolcüsüne (spell checker)(3), reklam sunucusuna (ad server)(4) ve indeks sunucularına (indeks servers)(5) paralel olarak sorguyu gönderir. Bu indeks sunucuları internetteki her kelime için bir girdi içerirler. Her bir girdi de bu girdiyi içeren dokümanları (web sayfaları, pdf dosyaları, power point sunumları, vs) sayfa derecelendirmeyle (page rank) sıralanmış bir şekilde listeler. Sayfa derecelendirmesi de karmaşık ve gizli bir formül ile belirlenir; ama o sayfaya yapılan bağlantılar (links) ve kendi derecelendirmesi o sayfasın yüksek sayfa derecesi almasında büyük bir role sahiptir.

Yüksek performans almak ve paralel arama yapılabilmesi için indeks shard (çömlek kırığı) denilen parçalara ayrılmıştır. 1. shard indeksteki tüm kelimeleri, her biri n en yüksek derecelendirmiş doküman göstericisini takip edecek şekilde 2. shard da 1. shard'dan sonraki n adet en yüksek derecelendirmiş doküman göstericisini takip edecek şekilde içerir ve bu shard'lar ilerleyerek devam eder. Burada kavramsal olarak aynı kelimelere aynı dokümanı işaret edecek şekilde sahip olan en az bir aynı işlevli eş shard bulunur. Aynı kelimenin farklı sorgularda yer alıp sorgu hızını düşürmemesi için. Web büyüdükçe de bu shard'lar k adet kelimeyi (k = n / 2) içerecek şekilde bölünürler. İndeks sunucuları bu kelimeyi içeren n adet doküman göstericisi döndürürler. Her bir kelime için dönen doküman göstericileri birleştirilir. Birleştirme aranan her bir kelimeden dönen dosya göstericilerin aynı olanlarının daha önde çıkmasını sağlayan mekanizma. (Tam bilinmeyen, gizli olan kısım.) Sıralanan dosya göstericileri de doküman sunucularına gönderilir (7) ve geriye doküman veya doküman parçacıkları (snippet) döner(8). Dönen dokümanlar tekrar sıralanır. Muhtemel yazım hataları heceleme sunucularında alınır(9). İlgili reklemlar reklam sunucusundan alınıp sonuç sayfasına eklenir(10). Sonuç sayfası html formatına dönüştürülerek sorguyu yapan kullanıcıya geri yollanır(11).

Birçok şirket yüksek performanslı, yüksek güvenilirlikli ekipman satın alırken Google tam tersini yaptı, düşük performanslı, ucuz ekipmanlar satın aldı. Bunu da fiyat/performans oranını optimize etmek için yaptı. Fakat ucuz ekipmanlarda çok arıza meydana geldiği için Google yazılımını arızalardan zarar görmeyecek ve ekipman bağımsız olacak şekilde tasarladı. Google sahip olduğu kişisel bilgisayarların %25'i arızalanır. Bu arızaların sebebi en çok yazımsal (çözümü bilgisayarı yeniden başlatmak), diskler, güç kaynakları ve bellekler olarak sıralanır. Google'ın kişisel bilgisayarları raflarda raf başına 80 adet olarak konumlandırılmışlardır. Aşağıdaki resimde görülebildiği üzere her raf içinde 4 switch ve bir üst seviyede de sürekli (her halükarda) (redundant) çalışabilen 2 switch yer almaktadır. Bu sayede herhangi bir switchde oluşabilecek sorunun sistemi etkilemesi engellenmiş olmaktadır. Her raftaki 4 switch'den 2'si gelen sorgular için diğer 2'si de yedekleme için ayrılmıştır. Tipik bir Google kümesi 5120 kişisel bilgisayardan oluşur.



Tipik bir bilgisayar 120 Watt bir raf da 10 kW enerji harcar. Bir rafa bakım için ortalama 3 metrekareye ihtiyaç duyulur. Bu parametrelere göre metrekarede 3000kW enerjiye ihtiyaç duyulur. Google 3 anahtar şey öğrenmiştir:

- Bileşenlerde sorunlar oluşacak bu yüzden planını ona göre yap.
- Her şeyi cevap verebilmek ve hazır halde tutabilmek için kopyala. (Yedekle)
- Fiyat/Performans oranını her zaman optimize et.

8.4.3) Communication Software for Multicomputers (Çoklu Bilgisayarlar için İletişim Yazılımı)

Çoklu bilgisayar programlama özel yazılım gerektirir, genellikle de işlemlerarası iletişimi ve senkronizasyonu sağlamak (handling) için kütüphaneler gereklidir. Yazılımsal olarak düşünüldüğünde birçok kısım ya da paketler (packeges) hem

MPP'lerde hem de küme bilgisayarlarda çalışırlar. Mesajlarla anlaşan sistemler iki veya daha fazla birbirinde bağımsız çalışan işlemlere sahiptirler. Örneğin bir işlem veriyi üretirken diğeri o veriyi kullanıyor olabilir. Çoğu mesajla anlaşan sistemler iki ilkel komut sağlarlar (kullanırlar) yolla (send) ve al (receive); ama birkaç çeşit anlamları olabilir. Bunlardan 3 temel biçimi şunlardır:

- Senkron mesaj yollama (Synchronous message passing)
- Tampon bellekli mesaj yollama (Buffered message passing)
- Bloklanamaz mesaj yollama (Nonblacking message passing)

Senkron mesaj yollamada yollayıcı (sender) mesajı yollar yollanan bloğu bloke eder ve alıcı mesajı işleyip geri cevap dönene kadar bekler. Cevap geldikten sonra yollayıcı işine devam edebilir. Bu yöntemde tampon belleğe ihtiyaç duyulmaz ve en basit mesajlaşma yöntemidir.

Tampon bellekli mesaj yollamada Alıcı hazır oladan mesaj yollandığında mesaj tampon bellekte (ya da diskte) tutulur ta ki alıcı mesajı alıncaya kadar. Böylece tampon bellekli mesaj yollamada alıcı meşgul olsa da gönderici mesajı yolladıktan sonra işine devam edebilir. Ancak bu yöntemde alıcının göndericiden gelen mesajı aldığı garanti değildir.

Bloklanamaz mesaj yollamada yollayıcı çağrıyı (call) yaptıktan hemen sonra işini yapmasına devam edebilir. Bu kütüphane işletim sistemine boş olduğunda yapması için bir çağrı yapar ve işlenecek çağrıyı tampon belleğe koyar ve yollayıcı donanımsal olarak (hardly) bloklanır. Bu yöntemin dezavantajı yollanan mesajın ne zaman cevaplandığından haberdar olmaktır. Bu soruna üretilen çözümler şunlardır: bir havuz sistemi oluşturup ona sormak veya tampon bellek müsait olduğunda bir kesme (interrupt) oluşturmaktır.

MPI – Message Passing Interface (Mesaj Yollama Arayüzü)

Bu bölümde birçok çoklu bilgisayar sisteminde kullanılan popüler mesaj yollama sistemi MPI kısaca tartışılacaktır. Önceleri çoklu bilgisayarlar için en popüler iletişim paketi PVM (Parallel Virtual Machine) idi. Son yıllarda bu yerini çoğunlukla MPI'ya bıraktı. MPI PVM'den daha zengin ve aynı zamanda daha karmaşıktır. Orijinal MPI versiyonu MPI-1 1997'de yerini MPI-2'ye bıraktı. MPI-1 PVM'nin yaptığı gibi işlem oluşturmak ve yönetmekle ilgilenmez. MPI 4 temel kavrama dayanır: iletişimci (communicator), mesaj veri tipi, iletişim işlemi (communication operation), ve sanal

topolojiler (virtual topologies). Bir iletişimci işlem grubu artı bir bağlamdan (context) oluşur. Bir bağlam uygulamanın (execution) bir safhasını işaret eden bir etikettir. Mesajlar gönderilip alındığında, bağlam bir diğer alakasız mesaja müdahale etmek için kullanılabilir. Mesajlarda bir çok veri tipi kullanılabilir. Bu tipler karakter, tamsayı, kesirli sayı ve bunlardan türemiş tipler olabilirler.

MPI geniş bir iletişim işlemi kümesi destekler (set of communication operation). Basit bir MPI mesaj yollama kullanımı aşağıdaki gibidir.
MPI _Send(buffer, count, data_type, destination, tag, communicator)
Bu çağrı hedefe data_type tipinde count adedinde bir tampon bellek yollar. Tag (etiket) alıcının mesajı seçmesine yardımcı olur. Communcator da hedefin hangi işlem grubunda (process group) içinde olacağını anlatır. Bu mesaja karşılık mesaj da aşağıdaki gibidir.
MPI_Recv(&buffer, count, data_type, source_tag, communicator, &status)
Bu mesaj da alıcının belirli bir tipte belirli bir kaynaktan belirli bir etikette (tag) bir mesaj aradığını duyurur.

MPI 4 temel iletişim modu sunar. Mod 1 senkron (yollayıcı alıcının cevabını bekler), mod 2 tampon bellekli (yollayıcı alıcının cevabını belmez işine devam eder), mod 3 standart (uygulama bağımsız senkron veya tampon bellekli olabilir)), mod 4 hazır (ready).

MPI sanal topoloji denen işlemleri ağaç (tree), çember (ring), ızgara (grid) ve diğer topolojilerde dizebileceği bir kavrama da sahiptir. Sanal topoloji iletişim yollarını (communication pathes) isimlendirmeyi sağlayan bir dizidir (arrangement) ve bu iletişimi kolaylaştırır. MPI-2 dinamik işlem oluşturmayı, uzak bellek erişimini (remote memory Access), bloklanamaz ortak iletişimi (nonblocking collective communication), ölçeklenebilir I/O desteğini, gerçek zamanlı işlem çalıştırmayı, ve yeni bir çok özelliği destekler. Bilimsel komitelerde MPI ile PVM destekçileri arasında çeşitli kamplaşmalar olmuştur. PVM'yi destekleyenler. PVM'nin kolay öğrenilebilir ve basitçe kullanılabilirliğini savunurlarken MPI'yı destekleyenler MPI'nın ekstra birçok işi yapabildiğini, standardının ve dokümantasyonunun olduğunu savunmuşlardır. Galiba sonuçta MPI kazanmıştır.