

# Paralel Programlamaya Giriş

- 
- Eray Özkural
- 
- TÜBİTAK UEKAE
- [eray@pardus.org.tr](mailto:eray@pardus.org.tr)
- 
- Bilkent Bilgisayar Mühendisliği
- Bölümü Paralel Hesaplama Grubu
- [erayo@cs.bilkent.edu.tr](mailto:erayo@cs.bilkent.edu.tr)



**PARDUS**

ULUSAL İŞLETİM  
S İ S T E M İ

# Sunum Çerçevesi

- Beowulf nedir, tarihçe, mimarisi
- Pratikte Beowulf
- Paralel hesaplamadaki temel kavramlar
- Türkiye'de YBH (gerçekten!)
- MPI öğrenelim

# Beowulf Nedir?

- **Amaç: Yüksek Başarımlı Hesaplama**
- **Network of Workstations (distributed camiasından gelen bir kavram)**
- **COTS (Commodity Off The Shelf) components**
- **Referans: T. Sterling, D. Becker, D. Savarese, et al. "BEOWULF: A Parallel Workstation for Scientific Computation, " Proceedings of the 1995 International Conference on Parallel Processing (ICPP), August 1995, Vol. 1, pp. 11-14**
- **Ayrıca tabii ki: [beowulf.org](http://beowulf.org)**

# Beowulf nereden çıktı

- “Beowulf class supercomputer”
- Beowulf vs. Grendel
- Klasik olarak Cray'ın makinaları ve diğer süperbilgisayarlar özel tasarlanmış ve imal edilmiş chipler/devrelerle yapılıyordu
- Bu pahalı canavarlara karşı kahramanca mücadele eden sistemlere Beowulf dendi
- Class I supercomputer: bütün yazılım/donanım commodity
- Class II supercomputer: bazı özel parçalara sahip

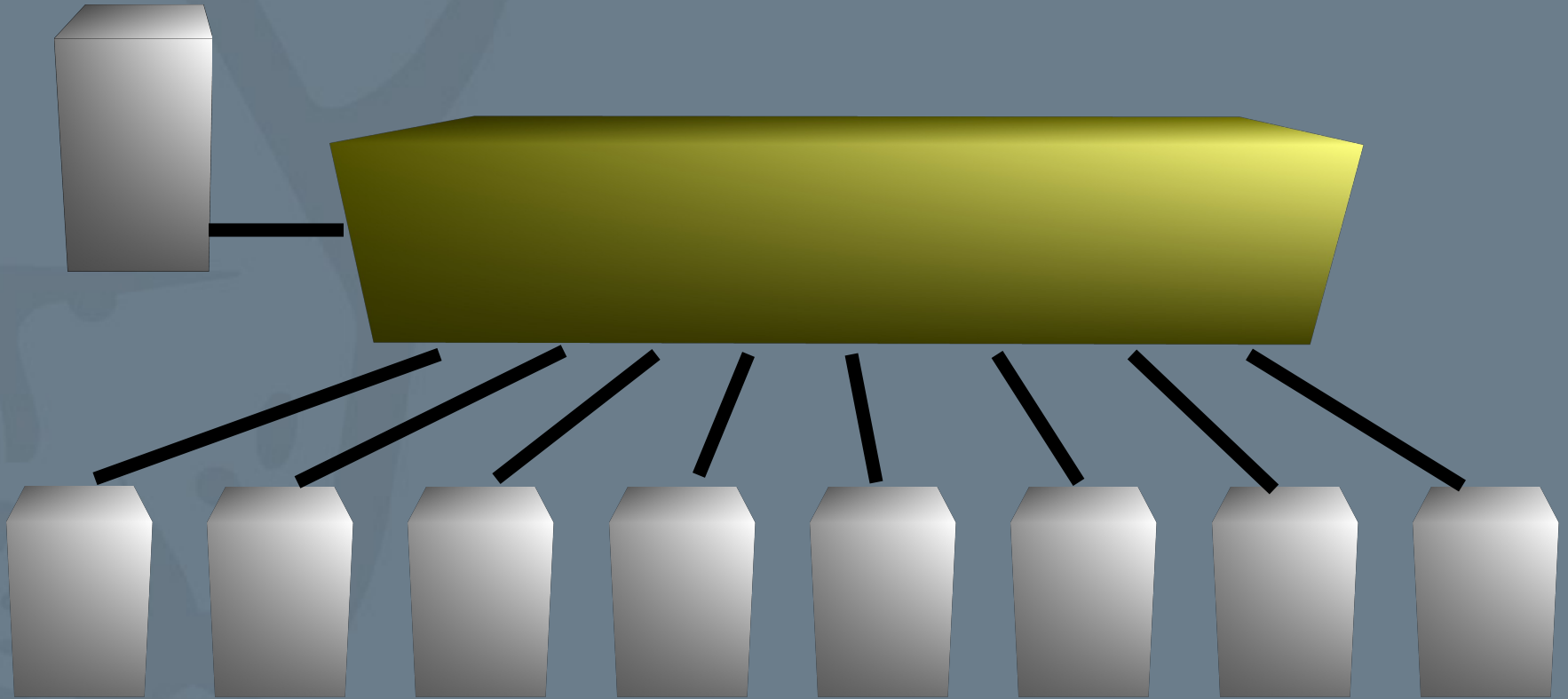
# Beowulf ilk uygulamalar

- **NASA, 1995 makalesi**
- **Computational Fluid Dynamics uygulaması**
  - tipik bir paralel uygulama
- **“Beowulf parallel workstation” 1993/1995**
  - 16 Intel DX4 CPU'lu anakart
  - işlemci başına 16 Mbyte DRAM
  - her işlemcide hard disk/controller
  - işlemci başına 2 tane 10 Mbit Ethernet
  - 2 video kartı, 2 monitör, 1 klavye

# Beowulf mimarisi (Donanım)

- **Bir arayüz düğümü**
- **n tane hesap düğümü**
  - Düğümler ethernet ile (1000Mbit, 100 Mbit) switch'e bağlanır
- **Bir arabağ ağı**
  - Üzerinde aynı anda bağımsız  $n/2$  tam kapasiteli konuşmaya izin verecek kapasitede bir switch
    - Bu durumda congestion sağlamayan ucuz ve yüksek performanslı switch'ler vardır

# Beowulf Mimarisi (donanım)



# Beowulf Mimarisi (Yazılım)

- **İşletim sistemi**

- Açık kaynak kodlu tercihan UNIX işletim sistemi: Linux, BSD.
- Kümeleme için gerekli ağ yazılımları (rsh, NFS, NIS, vs.)

- **Paralel programlama**

- Mesaj geçme yazılımı (MPI, PVM, vs.)
- Paralel uygulama kütüphaneleri
  - Paralel BLAS implementationları
  - CFD, vs. kütüphaneleri



- **Önce donanım planlaması yapmak gerekir**
  - Doğru donanım seçimi uygulamaya dayanır
  - Bir takım sorular sorarak başlayabiliriz
    - Paralel girdi / çıktı yapılacak mı?
    - Hesaplama mı ağır basıyor haberleşme mi?
    - Network latency ne kadar önemli?
    - Düğümlerin CPU'su hafızası ne olmalı?
    - Ne kadar para harcayabiliriz?
    - Verdiğimiz paranın karşılığını alabiliyor muyuz?

# Beowulf Yazılımı

- İyi bir linux dağıtımı seçelim
- Arayüz düğümünde geliştirme araçları, masaüstü, user accountları vs. yer alacak
- Hesaplama düğümlerinde temel sistem araçları ve paralel programı çalıştırmak için gerekli kütüphaneler yeterli
- Düğümleri özel bir IP ağına yerleştirelim
- Kümeleme yazılımı
  - Dosya paylaşımı (NFS)
  - Kullanıcı hesaplarının paylaşımı (NIS)
  - Düğümlere erişmek için ssh/rsh yapılandırması

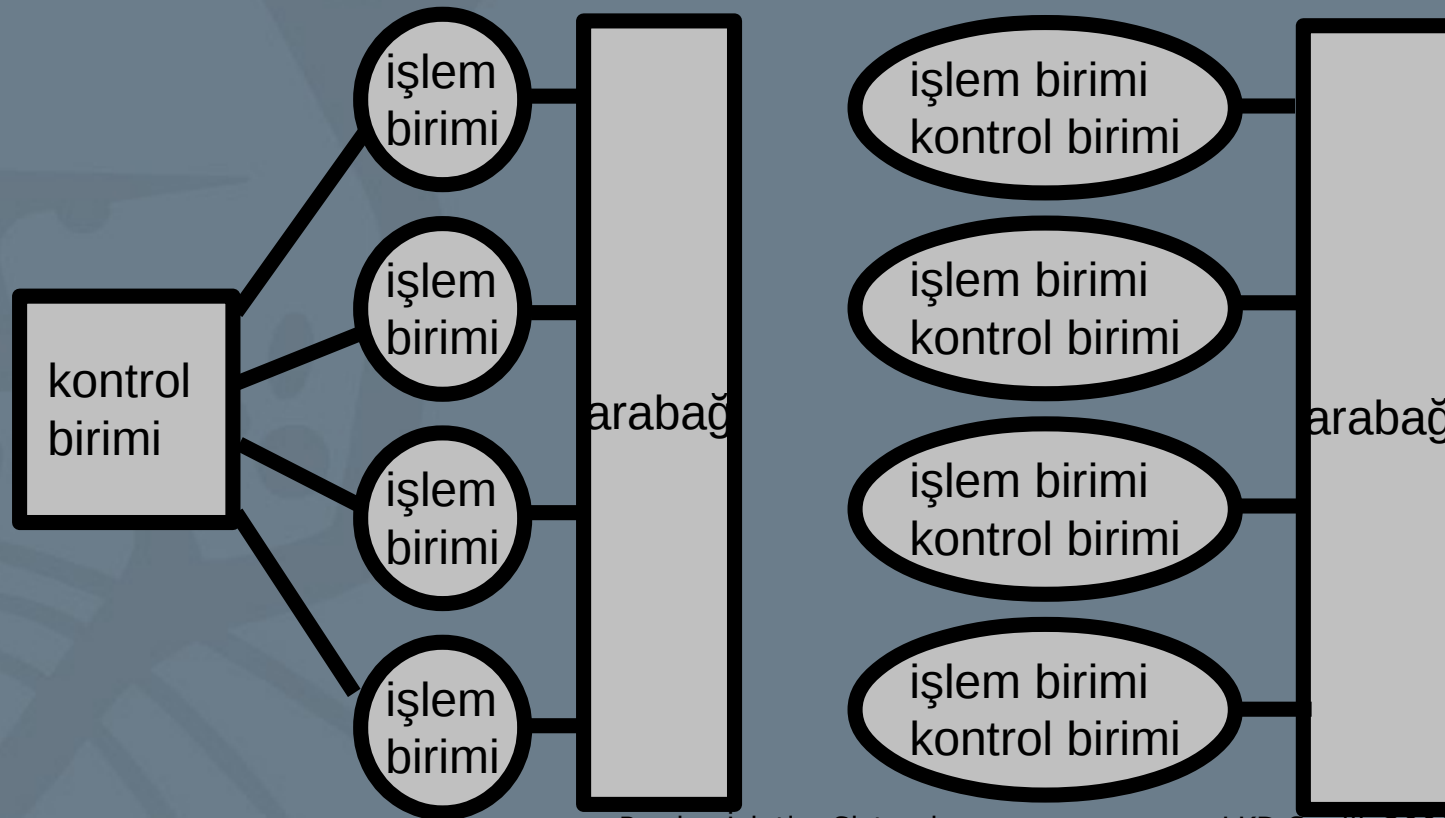
- **İlk kurduğum gerçek beowulf**
  - 2000/2001 yılları arasında
- **Donanım:**
  - 32 Pentium II 400 Mhz işlemci
  - node başına 128MB hafıza
  - 100Mbit ethernet network / SuperStack II switch
- **Yazılım**
  - Debian Linux
  - FAI (Fully Automatic Installation)
  - LAM/MPI, Charm++

- 2005 yılında TÜBİTAK paralel ağ sürüngei projesi için kurduk
- Donanım:
  - 48 Pentium IV 3.0Ghz
  - Düğüm başına 512MB hafıza 80GB disk
  - 1Gbit ethernet ve 100Mbit ethernet network'ü
  - Birisi internet'e erişim birisi düğümler arası erişim için
- Yazılım:
  - Mandrake mi Mandriva mı ne ondan
  - Warewolf (diskless boot)

# Paralel Hesaplama (Mimariler)

- **Kontrol mekanizması**

- SIMD (single instruction multiple data stream)
- MIMD (multiple instruction multiple data stream)



# Paralel Hesaplama (Mimariler)

- **Adres Uzayı Organizasyonu**

- Mesaj geçme (dağıtık)
- Paylaşılmış Adres Uzayı (UMA, NUMA)

- **Bağlama Ağları**

- Statik (doğrudan)
- Dinamik (routing var)
  - içinde switching elementler olan (2x2 crossbar gibi) switchler var, bunlar açılıp kapanıp gidiyor yerine

- **İşlemci gözenekleri**

- İnce gözenekli (fine-grain) (64K slow 1-bit procs)
- Kaba gözenekli (coarse-grain) (16 fast processors)

# Paralel Hesaplama (Teorik Modeller)

- **İdeal paralel bilgisayar PRAM adres paylaşımıdır:**
  - Exclusive-read, exclusive write (EREW) PRAM
  - Concurrent-read, exclusive write (CREW) PRAM
  - Exclusive-read, concurrent-write (ERCW) PRAM
  - Concurrent-read, concurrent-write (CRCW) PRAM

- **Dinamik arabağlar:**

- switching networks
- bus
- çok aşamalı arabağ ağları (Omega)
  - 2x2 switchleri birleştirmenin akıllı bir yolu
  - $p/2 \times \log p$  tane SE, Theta( $p \log p$ )

- **Durağan (statik) arabağlar: kac tel var?**

- clique: Theta( $p^2$ )
- star: Theta( $p$ )
- zincir ve halka: Theta( $p$ )
- mesh: Theta( $p$ )
- hypercube: Theta( $p \log p$ )



- **Durağan ağlar nasıl analiz edilir?**

- Çap: iki düğüm arasındaki en büyük uzaklık
- Connectivity: iki adam arasındaki min. yol sayısı
- Bisection Width, Bisection Bandwidth
  - Ağı ikiye ayırdık diyelim (iletişimi kopardık)
  - Bisection width: bunu yapmak kaç tane bağı çıkarmak gerekiyor
  - Bisection bandwidth: Bu linklerden saniyede kaç link geçiyor?

- **Başarım ölçüleri**

- Paralel çalışma zamanı  $T_p$
- Hızlanma (speedup)  $s = T_s / T_p$
- Etkinlik (efficiency)  $E = s / p$
- Bedel (cost, work)  $W = T_p \cdot p$ 
  - cost – optimal  $E = \Theta(1)$  (sabit yani)

- **Ölçeklenebilirlik (scalability):**

- Hızlandırmayı işlemci sayısı ile orantılı biçimde artırma kapasitesinin bir ölçüsüdür

- **Parallel overhead**

- paralelleştirmenin bedeli
- Kaynakları
  - İşlemciler arası haberleşme
  - Yük dengesizliği
  - Fazladan hesaplar:
    - Bazı hesaplar farklı işlemcilerde olduğu için birden çok kere yapılabilir (FFT)
    - Bazen en iyi seri çözüm paralel yapılamaz, o zaman daha az iyi bir seri çözüm paralel yapılır

# Performans (devam)

- **Amdahl'in kanunu:**

- Eğer  $W$  büyüklüğündeki bir problemin  $W_s$  lik seri bir kısmı varsa hızlanma en fazla  $W/W_s$  olur.

- **Bugünlük bu kadar yeter!**