

***BİL641***

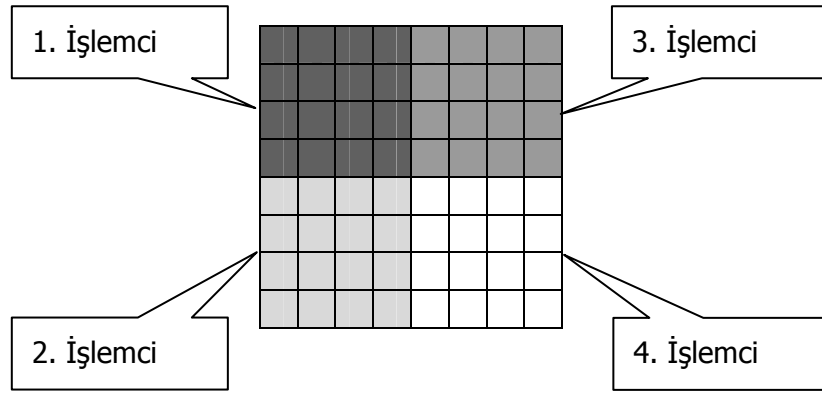
***MPI ve Canon Yöntemi Kullanarak  
Matris Çarpımı***

## 1. Uygulamanın Gerçekleştirimi:

Uygulamanın gerçekleştirimi esnasında standart Unix kütüphanelerinin kullanılmasına özen gösterilmiştir. Bu sayede geliştirilmiş olan uygulama tüm Unix ailesi işletim sistemleri altında çalışabilir olmuştur. Geliştirilmiş olan uygulama AIX, SunOS ve Linux türevleri altında test edilmiş ve başarı ile çalıştığı görülmüştür.

Canon matris çarpımı algoritması büyük olan mevcut matris'in küçük parçalara bölünerek çarpma işlemlerinin paralel olarak yürütülebilmesini sağlayan bir yöntemdir. Fakat Canon yönteminin bir handikapı çok fazla sayıda broadcast mesajı kullanmasıdır. Dolayısı ile çarpım işleminde işin ve mesajların dağıtılması çarpma işleminin kendisinden daha fazla vakit almaktadır.

Bu handikapı bir nebze olsun aşabilmek adına Canon yönteminin mesaj yayma mekanizması üzerinde ufak bir değişiklik yapılmış ve her işlemciye tüm matris'in yalnızca  $\frac{3}{4}$ 'ünün dağıtılması sağlanmıştır. Bu işlem başarıyı önemli ölçüde etkilemiş/iyileştirmiştir.



Yukarıdaki örnek dağılım detaylı olarak incelenirse 1. işlemcinin 4. işlemciye ait veriler ile "ASLA" ilgilenmeyeceği görülür. Dolayısı ile veri iletişimi yukarıda bahsettiğim yöntem ile optimize edilmiştir.

## 2. Test Ortamı:

Test ortamı Samsun Ondokuz Mayıs Üniversitesi Bilgi İşlem Daire Başkanlığı'nın oluşturmuş olduğu internet laboratuvarlarından birindeki 4 bilgisayarın kullanımı ile olmuştur. Bu bilgisayarların konfigürasyonları aşağıda verilmiştir.

<b>İşlemci</b>	:	Pentium 4 1.7Ghz
<b>Bellek</b>	:	256Mb

## 3. Test Koşulları:

10 farklı boyuttaki matrisler 4 farklı paralel işlem yöntemi ile test edilmiştir. En son olarak da 2048x2048 ebatlarındaki matris işletim sistemlerinin çekirdek katmanındaki çalışma zamanı değişkenleri değiştirilerek 4 işlemci, toplam 256 görev ile test edilmiştir (Normal koşullar altında

## MPI ve Canon Yöntemi Kullanarak Matris Çarpımı

Linux işletim sistemi bir görevin en fazla 50 alt görev (thread) yaratmasına izin vermektedir. Fakat çekirdek parametreleri oynanarak bu limit aşılabılır).

Her bir matris için 10 hesaplama yapılmış ve bu 10 hesaplamanın sonucu ortalama değer olarak elde edilmiştir.

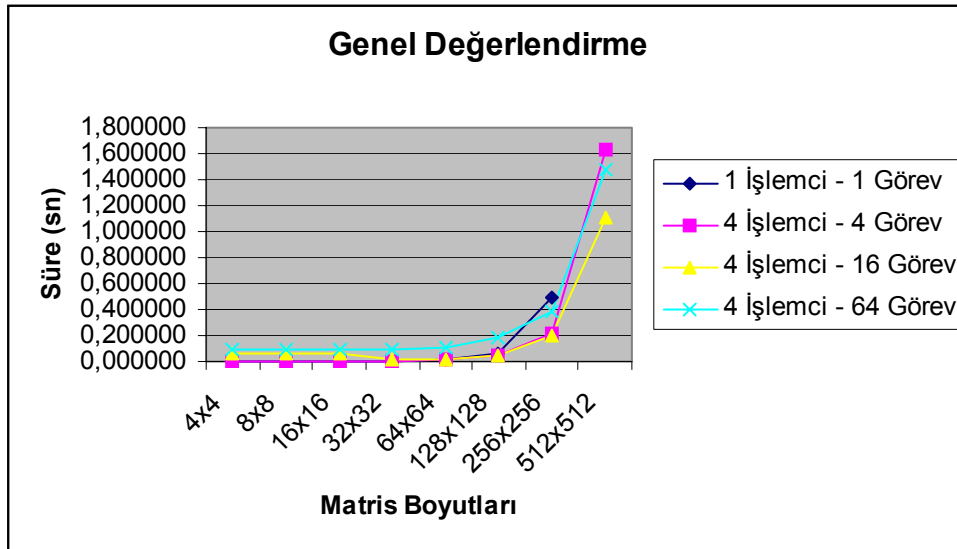
### 4. Test Sonuçları:

Her matris boyutu için rastgele tamsayı (integer) türünde veriler ile doldurulmuş 10 farklı matrisin çarpıtılması sonucu elde edilen zamanların ortalamaları alınmıştır. Zamanların ölçümü sırasında Unix ailesinde mevcut olan utime()<sup>1</sup> fonksiyonu kullanılmıştır. Tüm çalıştırma işlemlerinin sonucunda aşağıdaki beri tablosu elde edilmiştir.

	4x4	8x8	16x16	32x32	64x64	128x128	256x256	512x512	1024x1024	2048x2048
1G <sup>2</sup>	0,000026	0,000059	0,000223	0,001236	0,007721	0,056028	0,490503	17,447735	143,327248	1245,466443
4G <sup>3</sup>	0,000992	0,001064	0,001402	0,003497	0,010598	0,040877	0,214387	1,623153	35,888184	294,429315
16G	0,068482	0,067997	0,068114	0,012822	0,017414	0,044160	0,195307	1,105587	8,873468	243,374278
64G	0,095495	0,090688	0,093205	0,098320	0,106666	0,179104	0,388136	1,482625	7,932643	61,866652
256G	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	16,466663

### 5. Genel Değerlendirme

Yukarıdaki tablodan ve aşağıda verilmiş olan grafikte görebileceği üzere her paralel işlem yönteminin avantajlı olduğu durumlar sözkonusudur.



<sup>1</sup> Bu fonksiyon saniyenin 1/100000'i kadar hassastır.

<sup>2</sup> Tek işlemci ile

<sup>3</sup> Dört işlemci ile

## MPI ve Canon Yöntemi Kullanarak Matris Çarpımı

Örneğin 1 İşlemci/1 Görev seçeneği matris boyutu 128'i geçmediği sürece en etkili seçenek olmaktadır. Matris boyutu 128 için en uygun seçenek 4 İşlemci/4 Görev olmaktadır. Matris boyutu 256 ila 512 içinse 4 İşlemci/16 Görev en iyi sonucu vermektedir. 1024 ebatındaki bir matris için 4 işlemci/64 Görev en başarılı seçenektir.

İşlemci sayısının artması çoğu durumda yapılacak işi hızlandırmak yerine yavaşlatmaktadır. Örnek olarak 8x8'lik matrisi inceleyecek olursak, dört işlemci ile bu matrisin çarpımı tek işlemci ile yapılan sürenin yaklaşık 20 katı kadar sürede yapılabilmektedir.

