



PARALEL BİLGİSAYAR

MATRİS ÇARPMA UYGULAMASI

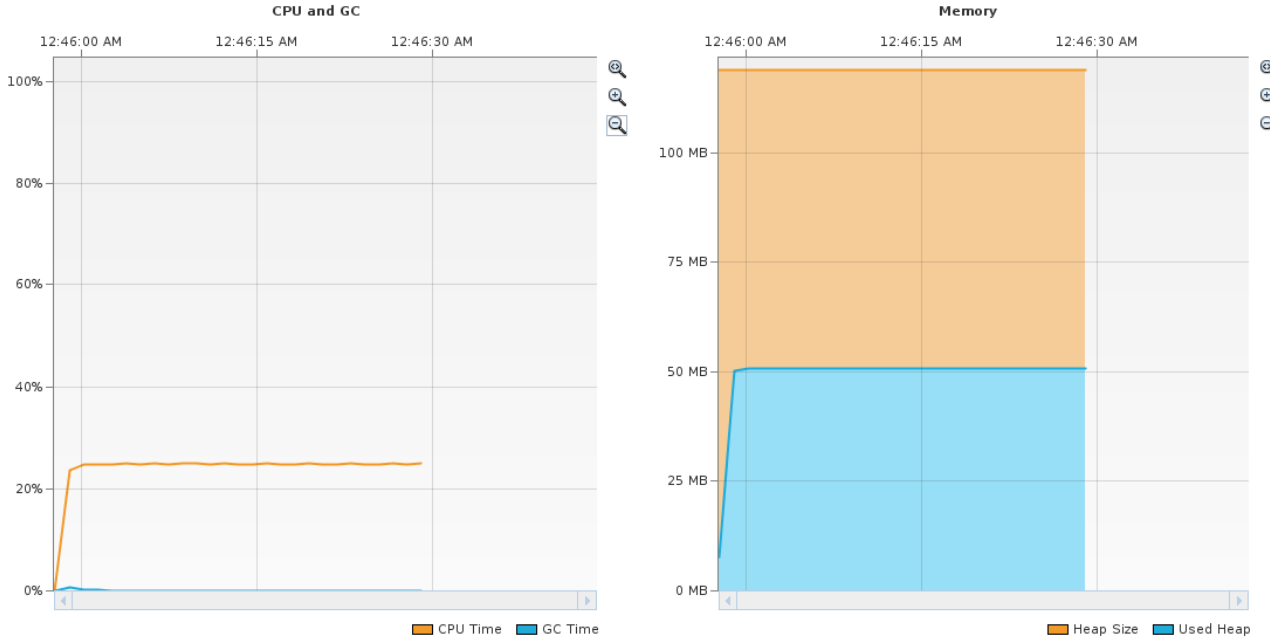


DERSİN SORUMLUSU : YRD.DOÇ.DR İBRAHİM SAVRAN

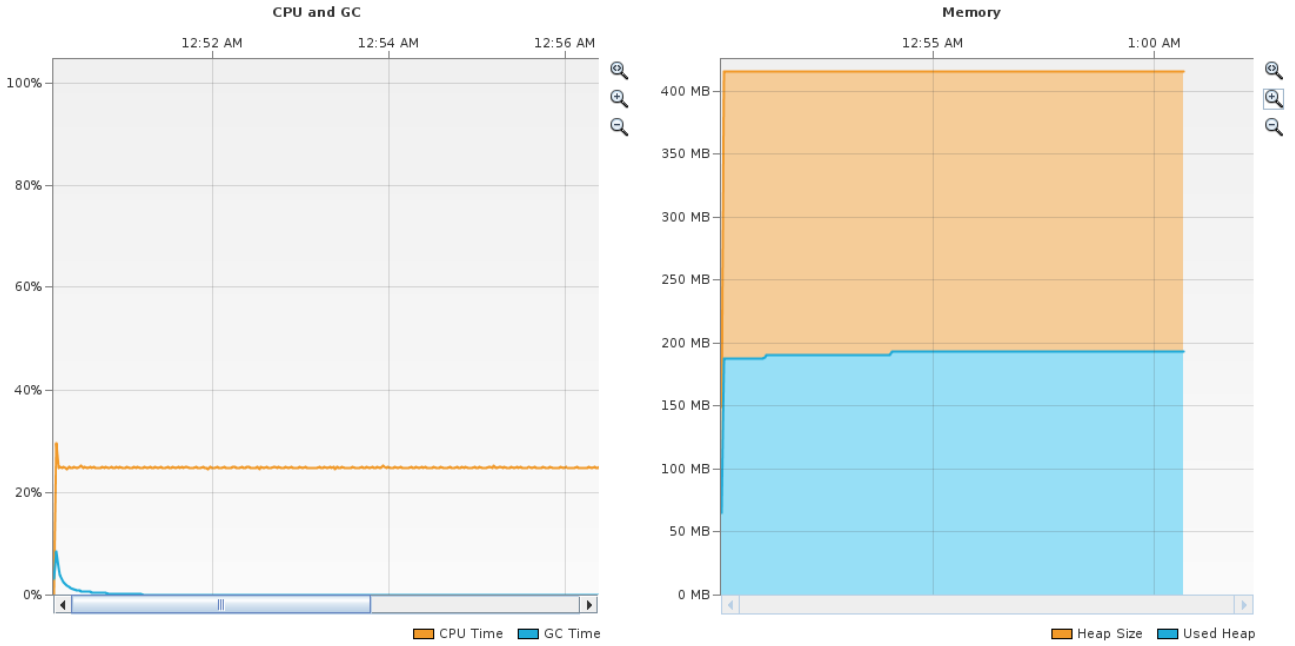
HAZIRLAYANLAR

ENDER YILDIRIM	295074
MURAT CAN VARER	295083
TULİN YARIMCI	295079

1)NORMAL MATRİS ÇARPIMI



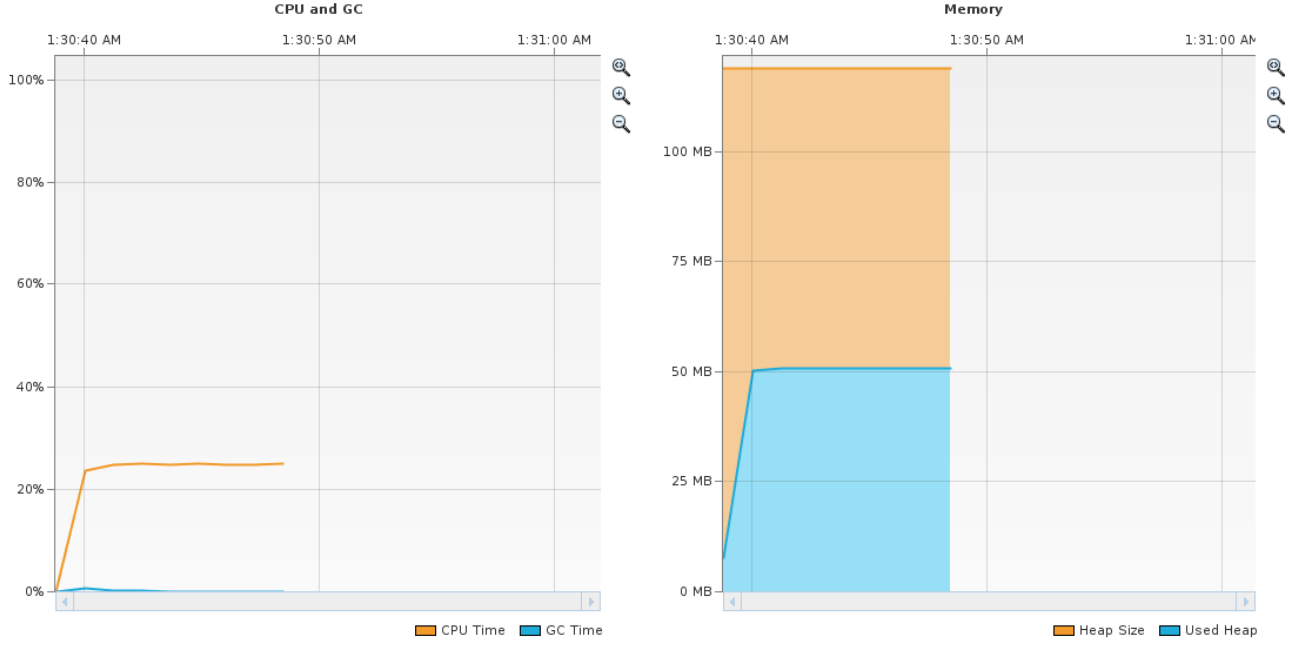
Şekil1.1 :2000x2000 matris çarpımı memory ve cpu kullanımı



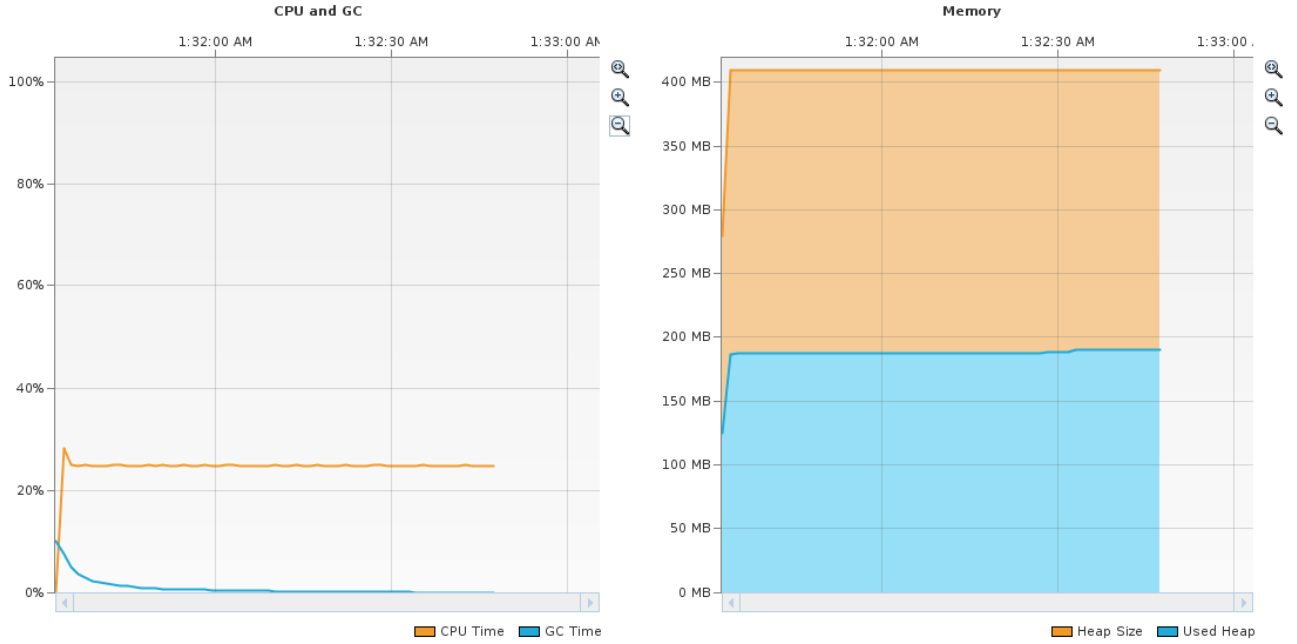
Şekil1.2 : 4000x4000 matris çarpımı memory ve cpu kullanımı

Şekildede görüldüğü gibi iki çarpımdada cpunun kullanma yüzdeleri bir anda artıyor ve sonra stabil bir şekilde ilerliyor.Fakat memorydeki tablolara bakarsak **Şekil1.1** 50MB gibi bir alan kullanırken **Şekil1.2** ortalama 190MB gibi bir alan kullanıyor.

2)TRANSPOZU ALINMIŞ GİBİ ÇARPMA



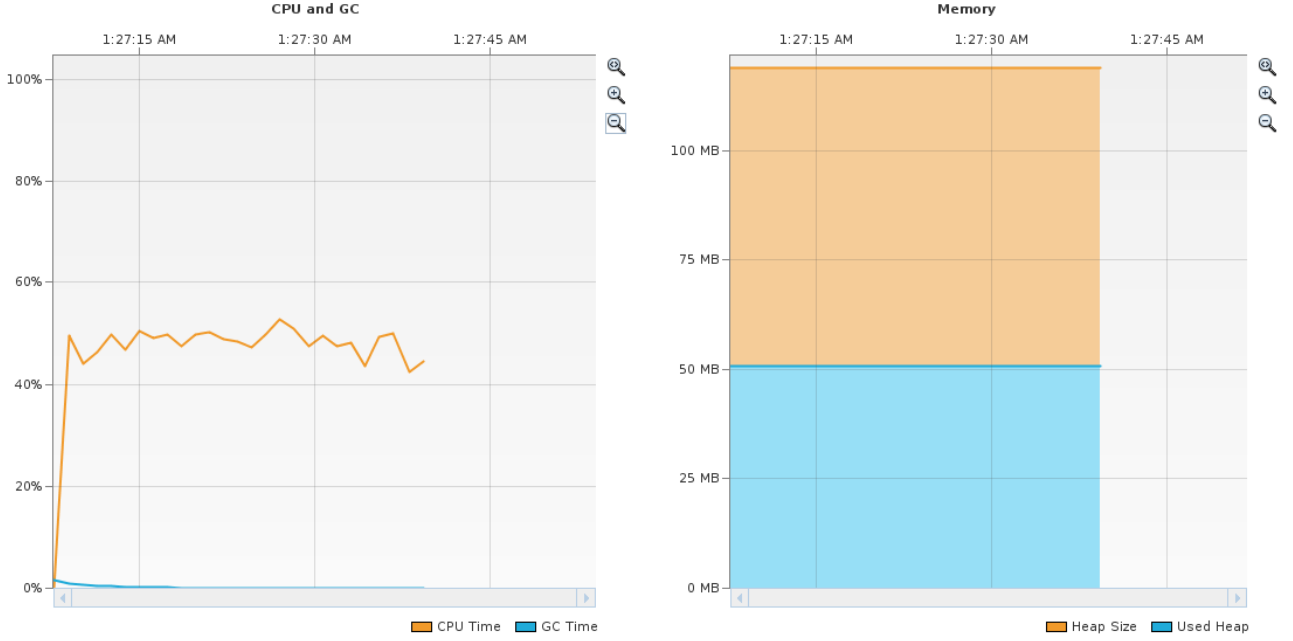
Şekil2.1 : 2000X2000 Transpozu alınmış gibi çarpma



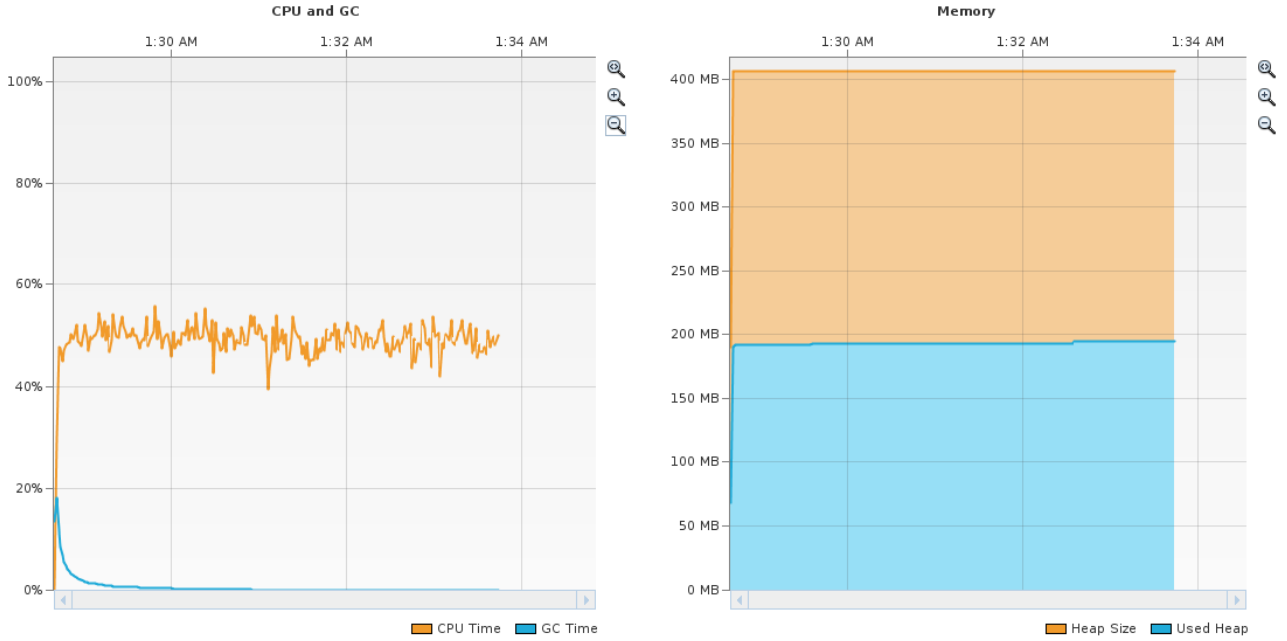
Şekil2.2 : 4000X4000 Transpozu alınmış gibi çarpma

Şekillerde de görüldüğü gibi Normal matris çarpım işleminde kullanılan CPU ve bellek miktarı aynı olmasına rağmen Transpozu alınmış gibi olan çarpım daha kısa sürede işlemini tamamlıyor.Neredeyse %50 zamandan kazanç var.Bunun sebebi satır'larda paralel işlemlerin yapılmasıdır.

3) 4 THREAD'li BLOKLARA BÖLÜNMÜŞ MATRİS ÇARPIMI



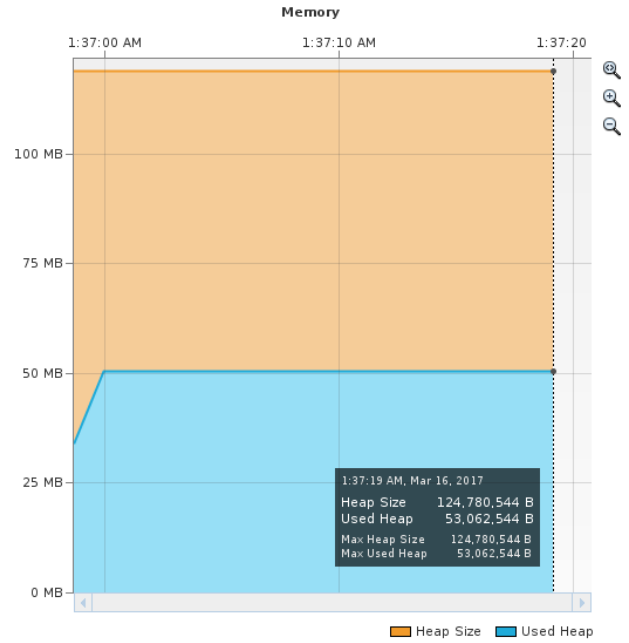
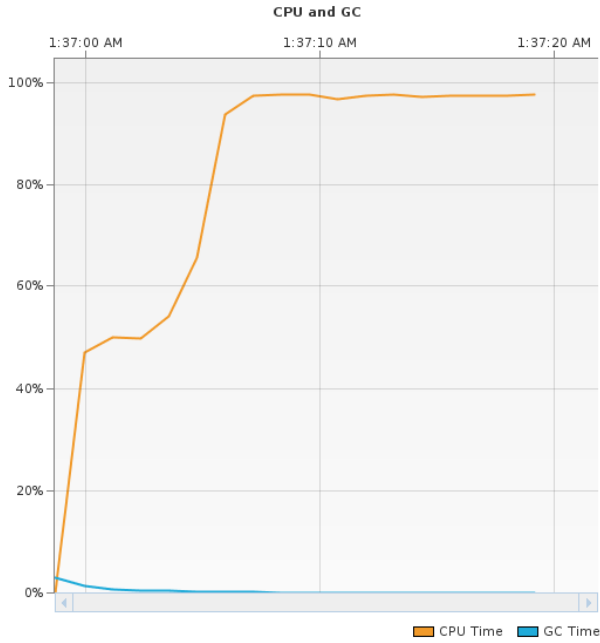
Şekil3.1 : 2000X2000 bloklu matris çarpımı



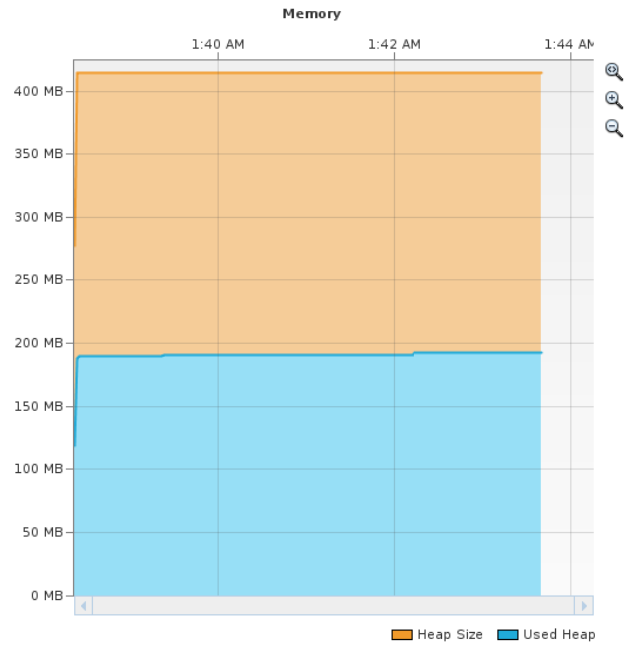
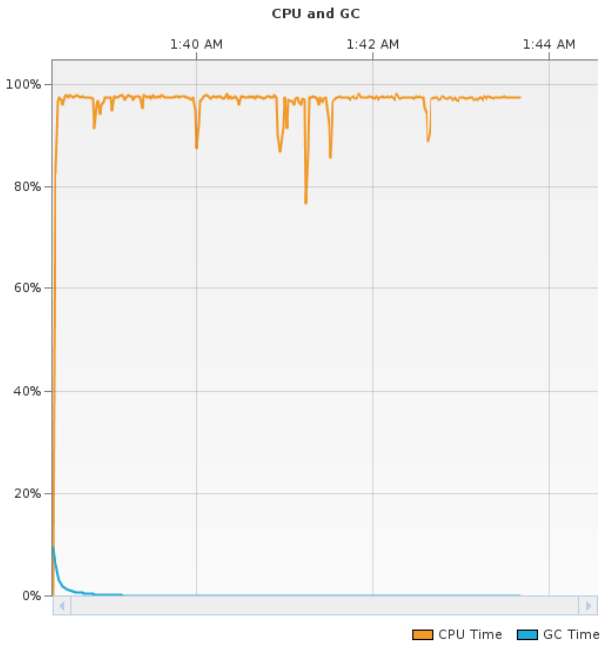
Şekil3.2 : 4000x4000 bloklu matris çarpımı

Şekillerde de görüldüğü her çarpımda da CPU'yu kullanımları değişiyor. Bunun sebebi threadlerin farklı kaynaklar tüketmesi. Memory kısmındada matris boyutu az olan daha az heap kullanırken boyutu fazla olan daha fazla heap kullanıyor.

4) 4 THREAD'li BLOKSUZ MATRİS ÇARPIMI



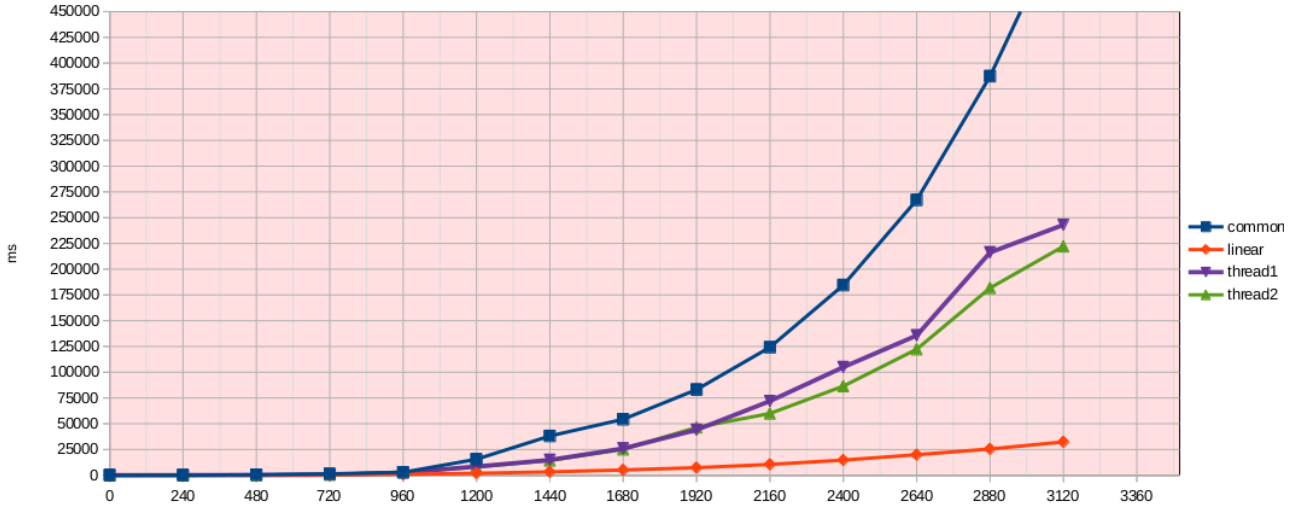
Şekil4.1 : 2000X2000 bloksuz Matris çarpımı



Şekil4.2 : 4000x4000 bloksuz Matris çarpımı

Şekillerde de görüldüğü bu yöntemde CPU'nun neredeyse tamamının kullandığını görmekteyiz. CPU'yu çok meşgul etmesi bir dezavantaj. CPU'nun bu kadar yüksek kullanımı daha kısa sürede yapılması beklenirken zaman almasının sebebi threadlerin aynı anda okuma/yazma yapamamasıdır.

SON OLARAK BU 4 KODLARA GÖRE ORTAK BİR GRAFİKTE ÇARPIM İÇİN GEÇEN SÜRELERE BAKALIM



BUILD SUCCESSFUL (total time: 48 minutes 37 seconds)

Mavi ile gösterilen(**Common**) normal satırXsütün çarpmasıdır.

Turuncu ile gösterilen(**Linear**) satırXsatır şekilde transpozu alınmış gibi çarpma şekildir.

Mor ile belirtilen şekil (**Thread1**) threadleri bloklara bölünmüş hali şeklindedir,yani her thread'e ilk başta işlem yapacağı bloklar veriyor işi biten thread sonlıyor ve diğer thread devam ediyor.Biz 4 thread'li yaptık

Yeşil ile belirtilen şekil(**Thread2**) ilk olarak threadler oluşturuyor kaynağa random erişiyorlar burada işletim sisteminin bir schedule olayı söz konusu.

SIZE ile gösterilen kaç kaç matris olduğu örnk: 720x720

MS ile gösterilen kaç milisaniyede tamamlandığı.