

# Çok Çekirdekli Sistemlerin Yazılım Kalitesi Üzerine Etkileri

**Tolga OVATMAN, Feza BUZLUCA**

**İstanbul Teknik Üniversitesi  
Bilgisayar Mühendisliği Bölümü**

## Problem:

- Donanımdaki değişim: Çok çekirdekli yapılar
- Yazılım tasarımının dikkate aldığı konular farklı:
  - Esneklik, tekrar kullanılabilirlik vs.

Özellikle çekirdek sayısı arttıkça verimlilik problemi artıyor.

## Soru:

- Donanımdaki değişim yazılımları nasıl etkileyecek?
- Mevcut yazılımlar çok çekirdekli yapılara nasıl uyarlanabilir?
- Yazılım tasarımı çok çekirdekli yapılar dikkate alınarak yapılabilir mi?

## Donanım:



- Tek kırımda çok çekirdek (işlemci)
- Her işlemcinin yerel belleği (1. düzey cep bellek)
- Ortak bir 2. düzey cep bellek
- Sıkı bağlı bir sistem

## Mevcut Çalışmalar:

- Bir kısmı nesneye dayalı değil: C, Fortran
- Çalışmalar konuyu tasarım düzeyinde değil kodlama düzeyinde ele alıyor.

## Yapılabilecekler:

1. Derleyici / ara katman yardımıyla mevcut yazılımların çok çekirdekli yapılara uyarlanması

Günümüz yazılımlarının kalıplar nedeniyle bir karakteristiği var mı?

- Örneğin fazla bağımlı olan nesnelere (coupling) aynı çekirdekte mi yer almalı?
- Denetçi, proxy gibi nesnelere bölünmeli mi, kopyaları mı çıkartılmalı?
- Strateji nesnelere nasıl yerleştirilmeli? ...

## Yapılabilecekler:

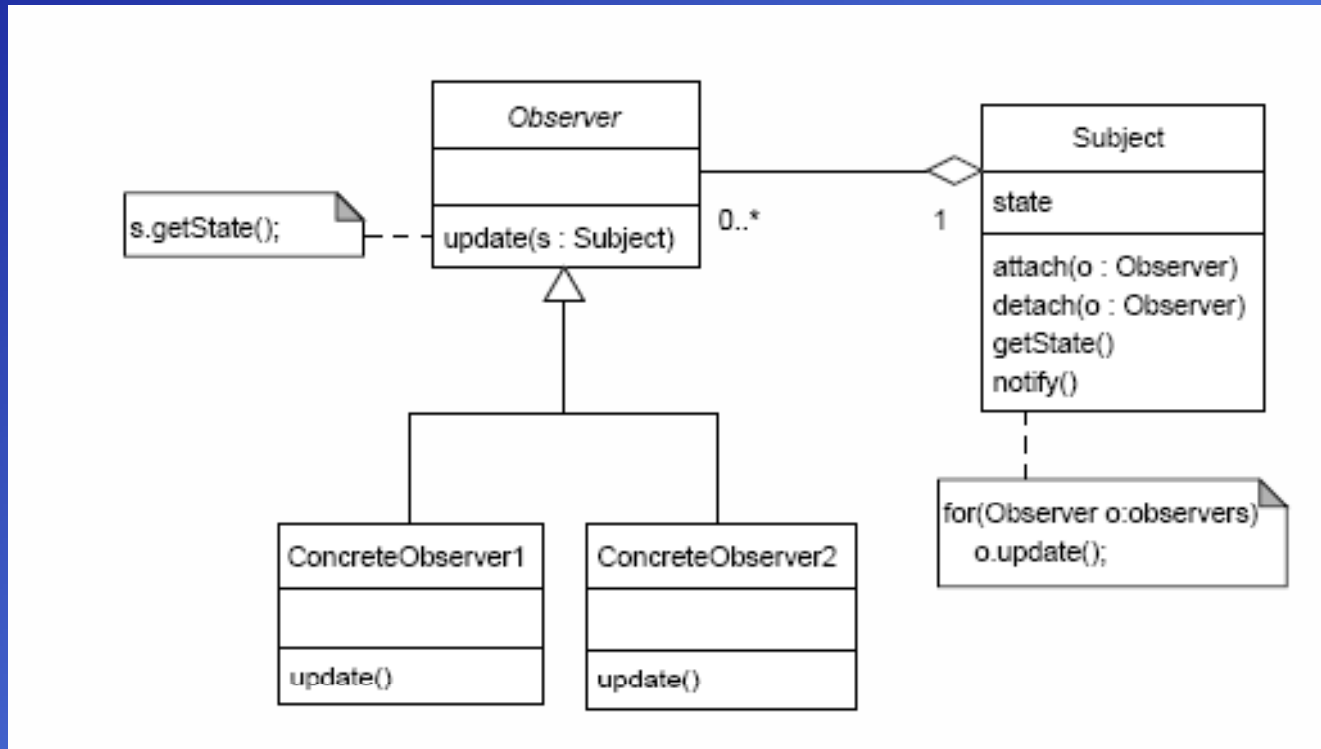
### 2. Tasarımcının yapabilecekleri

- Tasarımcı donanımdan ne kadar haberdar olmalı? Kazanç?
- Yeni kalıplar?
- UML'e eklentiler

## Kalıpların çok çekirdekli sistemlerde incelenmesi:

Gözlemci (observer) kalıbı:

Sadece okuma yapıldığı varsayılıyor.

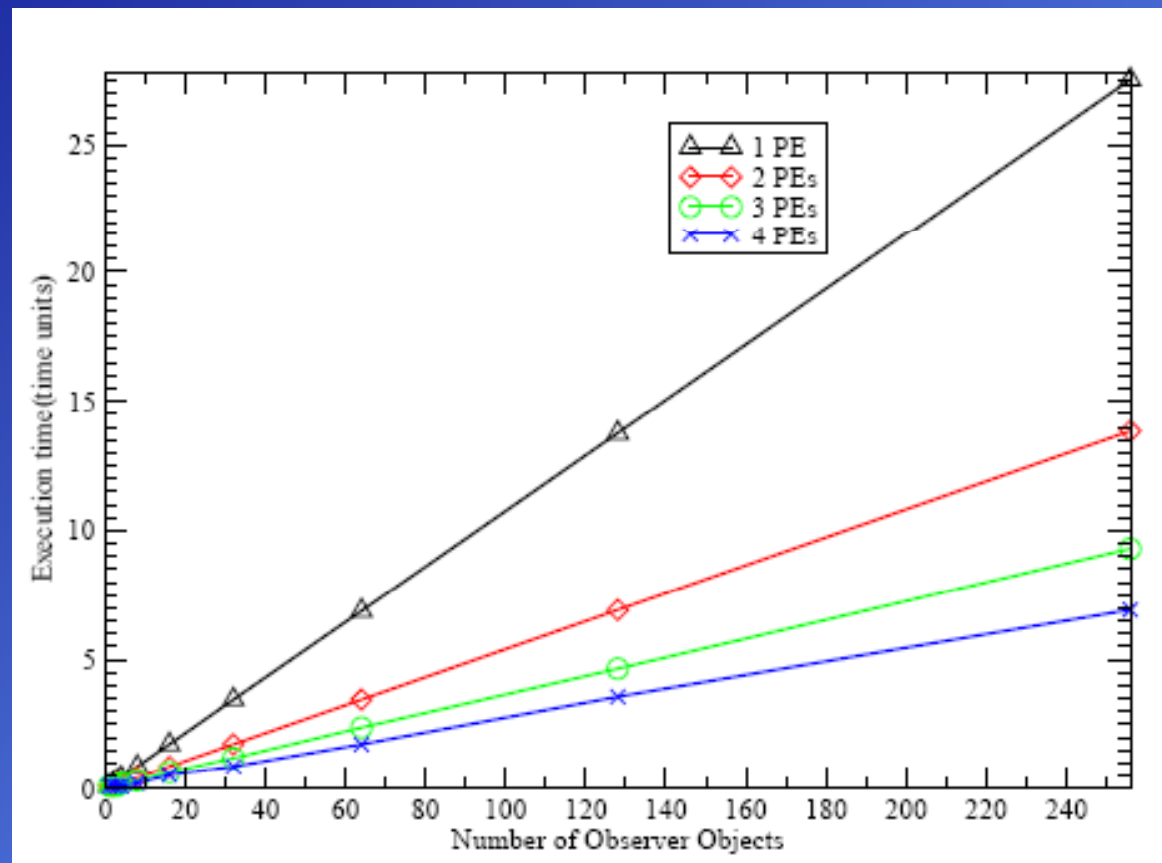


### Deney ortamı:

- 4 İşlemci
- Linux 2.4
- C++
- Pthread kütüphanesi

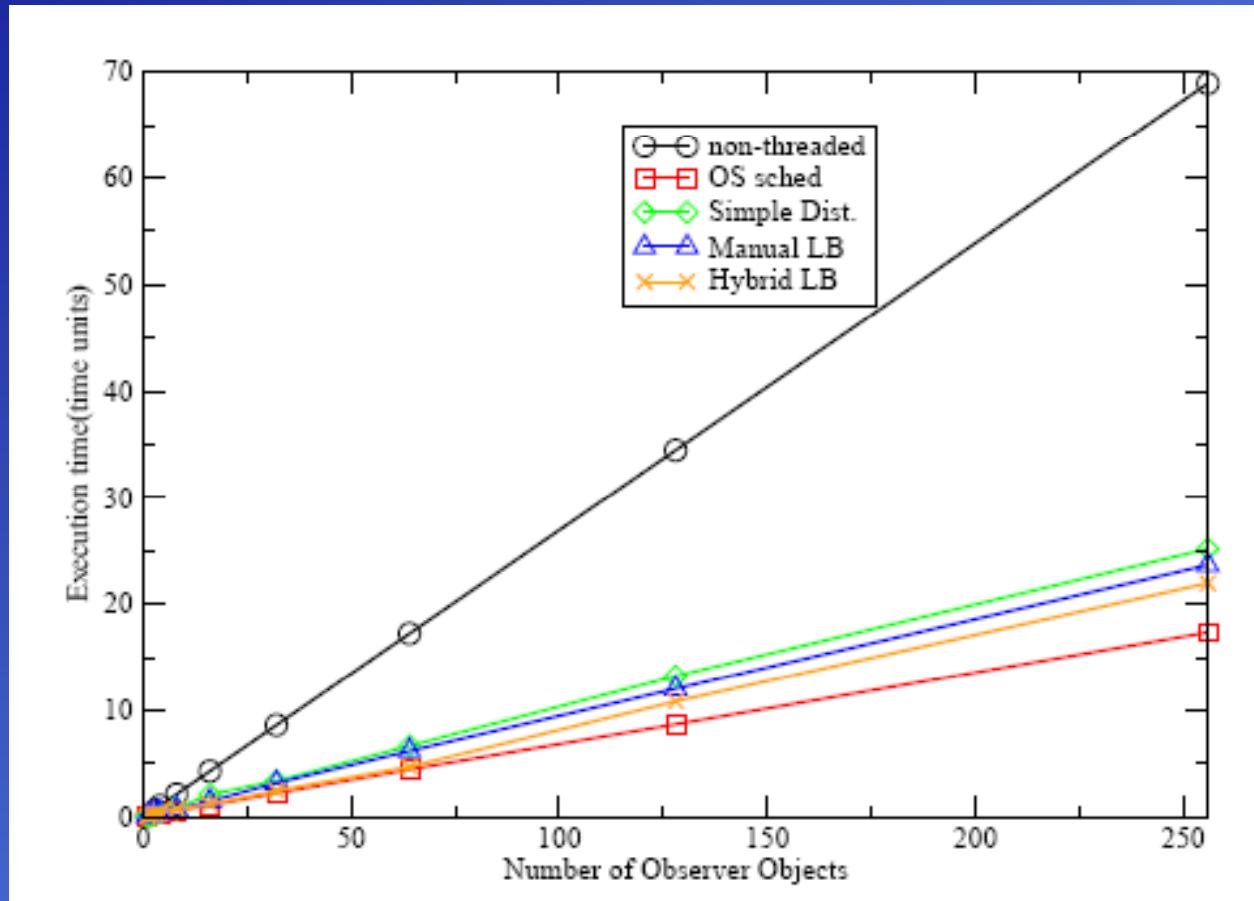
Önce eşit yüklü gözlemci nesneleri ardından farklı yüklerdeki gözlemci nesneleri sistemde çalıştırılmıştır.

## Eşit yüklü gözlemciler:





## Farklı yükte gözlemciler:



En "kötü" dağıtım bile seri çalışmadan daha iyi sonuç veriyor.

## Sonuçlar:

- Donanımdaki deęişim yazılım kalitesi üzerinde etkili olacaktır.
- Çok çekirdekli yapıları da dikkate alarak yazılım kalitesini arttırmak için farklı düzeylerde çalışmalar yapılabilir:
  - Ara katman
  - Programlama
  - Tasarım, modelleme
- Yeni tasarım kalıpları oluşturulabilir.
- Yeni kalite kriterleri ve metrikler gereklidir.