Overhead at Runtime (JIT Performance Issues)

إيه هي مشكلة الـ Overhead دي؟

- زي ما شرحنا قبل كده، الـ JIT بيترجم الكود من IL لـ Machine Code **وقت التشغيل**
- المشكلة إن عملية الترجمة دي **بتاخد وقت**، وده بيخلي البرنامج **أبطأ في البداية**
- تخيل إنك بتفتح برنامج كبير زي Visual Studio، الـ JIT هيبقى مشغول يترجم الكود أول ما تبدأ تستخدم أي حاجة فيه

مثال يوضح المشكلة:

```
public void CalculateComplexMath()
{
    // كود معقد جداً - 1000 سطر مثلاً
    for(int i = 0; i < 1000000; i++)
    {
        // Complex calculations...
    }
}
```

- أول مرة المستخدم هينادي على Function دي، الـ JIT هيقعد **يترجمها الأول** قبل ما ينفذها
- وده هيخلي المستخدم **يستنى** شوية قبل ما يشوف النتيجة

الحلول اللي Microsoft عملتها عشان تقلل الـ Overhead

1. Method Caching

- لما الـ JIT يترجم Method معينة، **بيحفظها في الذاكرة**
- فلو المستخدم نادى على نفس الـ Method تاني، **مش هيترجمها مرة تانية**
- هيستخدم الـ Machine Code اللي موجود في الذاكرة علطول

```
static void Main()
{
    CalculateSomething();  // ← هنا الـ JIT (بطيء) هيترجمها
    CalculateSomething();  // ← هنا هيستخدم المترجم خلاص (سريع!)
    CalculateSomething();  // ← برده سريع!
}
```

2. Ahead-Of-Time Compilation (AOT)

- Microsoft عملت حاجة اسمها **ReadyToRun**
- دي بتخلي جزء من الكود **يتترجم مقدماً** وقت الـ Build
- فلما المستخدم يشغل البرنامج، جزء كبير منه **جاهز خلاص**

```
# Command عشان تعمل ReadyToRun بتاعك للمشروع
dotnet publish -c Release -r win-x64 --self-contained -p:PublishReadyToRun=true
```

النوع التاني: Native AOT (.NET 7+)

- ده أحدث حل من Microsoft
- بيترجم **البرنامج كله** لـ Machine Code قبل التوزيع
- النتيجة: **مفيش JIT overhead خالص!**

```
<!-- في ملف .csproj -->
<PropertyGroup>
    <PublishAot>true</PublishAot>
</PropertyGroup>
```