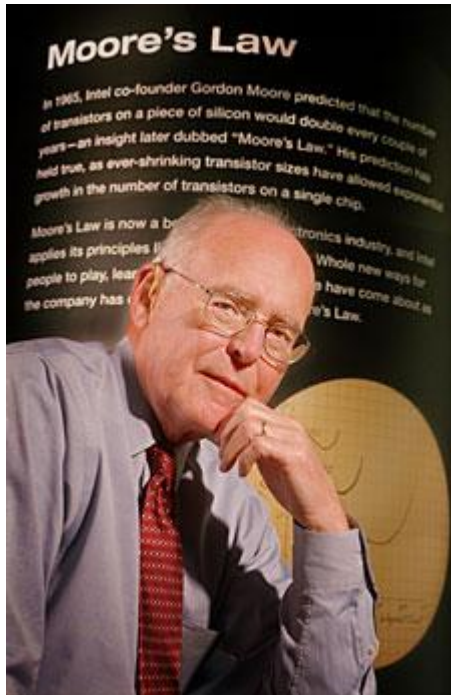


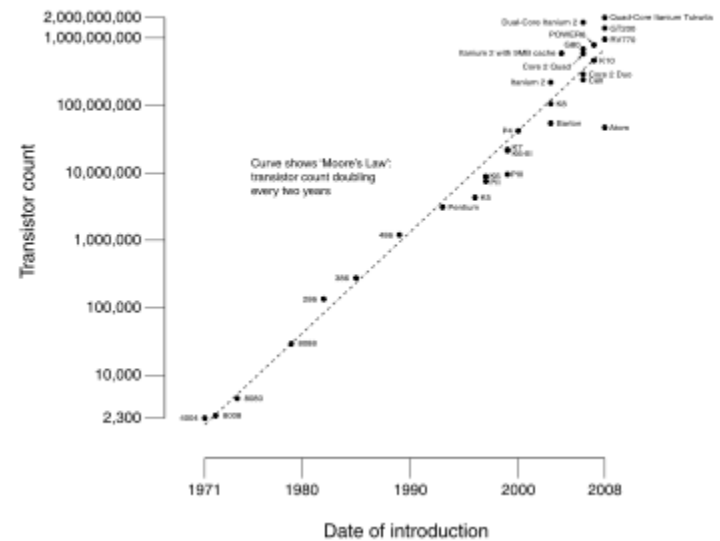
Parallell programmering med .NET

Pär Nordström
Valtech

Moore's Law



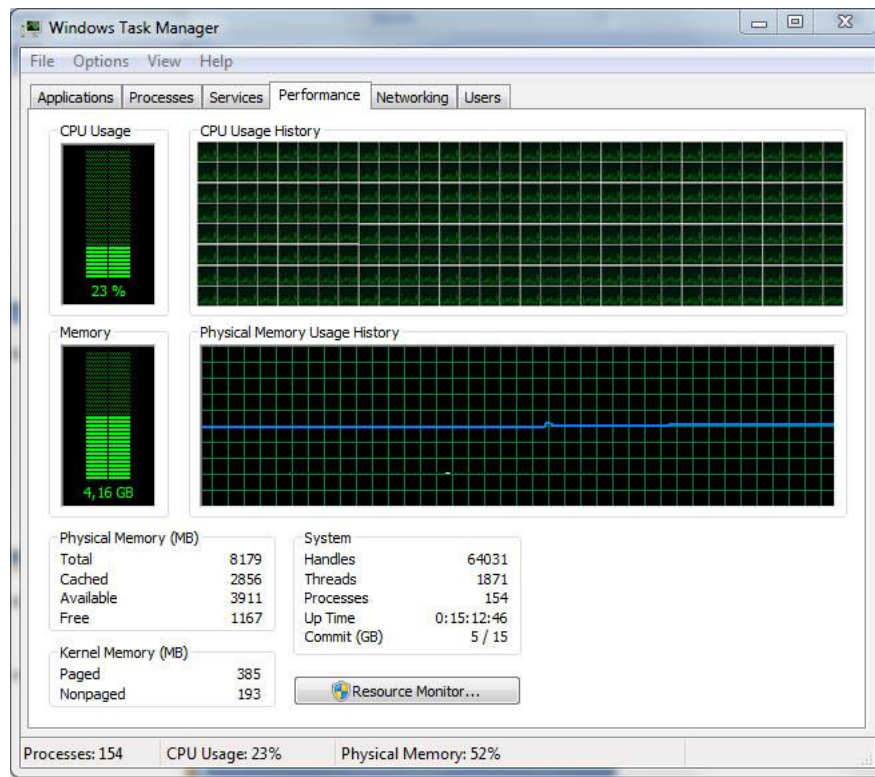
CPU Transistor Counts 1971-2008 & Moore's Law



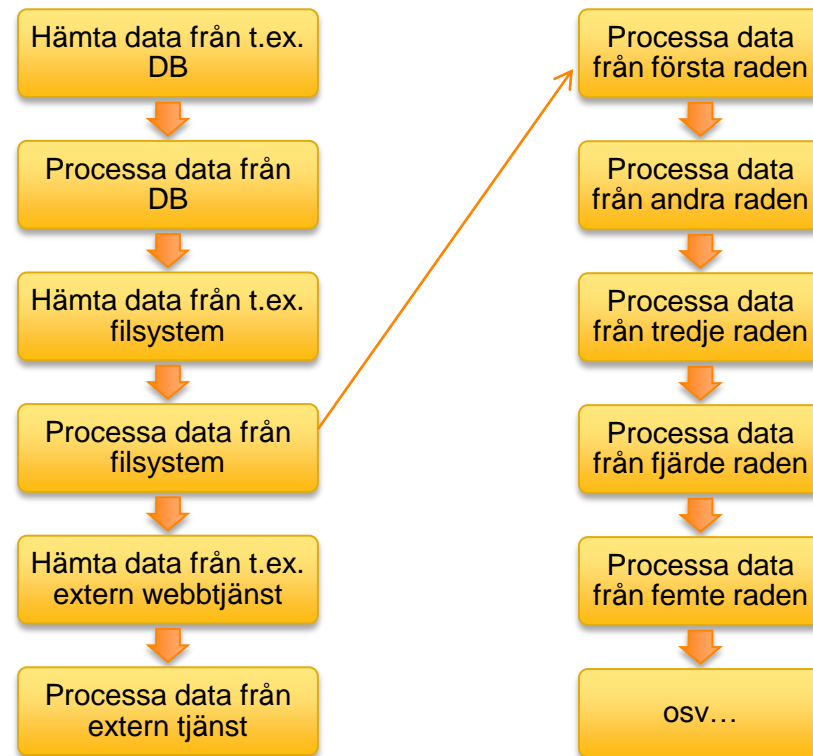
A Brick Wall For Serial Performance
=
Power Wall + Memory Wall + ILP Wall

David Patterson
Parallel Computing Laboratory,
University of California, Berkely

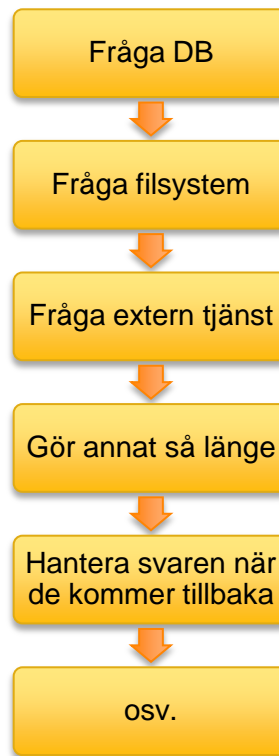
Fler kärnor per CPU!



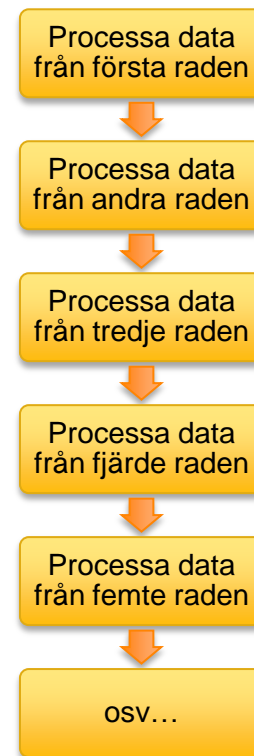
De flesta program är gjorda för seriell exekvering:



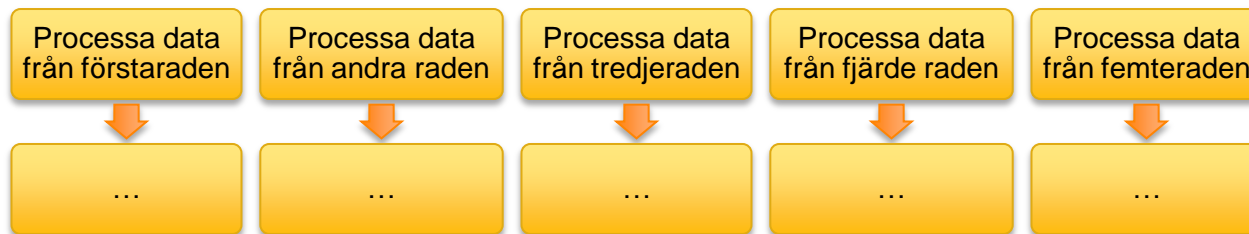
Lösning – Asynkrona anrop



Lösning – Parallell exekvering



Lösning – Parallell exekvering



```
static void MultiThreading()
{
    Thread t = new Thread(new ThreadStart(DoWork));
    t.Start();
}

static void DoWork()
{
    //...
}
```

Windows Task Manager

File Options View Help

Applications Processes Services Performance Networking Users

Image Name	User Name	CPU	CPU Time	Memory (...)	Threads	I/O P...
devenv.exe *32	par.nor...	01	00:04:47	238 592 K	59	2
devenv.exe *32	par.nor...	00	00:02:09	138 436 K	52	2
OUTLOOK.EXE *32	par.nor...	00	00:01:21	37 660 K	52	3
spotify.exe *32	par.nor...	00	00:08:28	15 132 K	51	1
devenv.exe *32	par.nor...	00	00:00:33	134 580 K	43	2
explorer.exe	par.nor...	00	00:10:48	36 376 K	35	39
taskmgr.exe	par.nor...	02	00:00:02	3 112 K	20	
sidebar.exe	par.nor...	00	00:00:05	6 908 K	20	
BTStackServer.exe	par.nor...	00	00:00:00	3 744 K	18	
POWERPNT.EXE *32	par.nor...	00	00:07:34	4 736 K	17	4
Dell.UCM.exe *32	par.nor...	00	00:00:11	7 340 K	15	
Dell.ControlPoint.exe	par.nor...	01	00:02:05	11 972 K	15	
SmcGui.exe	par.nor...	01	00:03:00	2 568 K	15	14
SearchProtocolHost.exe *32	par.nor...	00	00:00:00	1 856 K	13	
Personal.exe *32	par.nor...	00	00:00:10	1 552 K	13	14
csrss.exe		00	00:01:26	1 384 K	13	53
SnippingTool.exe	par.nor...	00	00:00:00	1 892 K	11	
Fsi.exe *32	par.nor...	00	00:00:02	14 108 K	11	
iTunesHelper.exe *32	par.nor...	00	00:00:00	1 664 K	11	
msvsmon.exe	par.nor...	00	00:00:00	2 088 K	10	
ccApp.exe *32	par.nor...	00	00:00:01	440 K	10	

Show processes from all users End Process

Processes: 122 CPU Usage: 20% Physical Memory: 34%

```
static void UsingBackgroundWorker()  
{  
    var bgw = new BackgroundWorker()  
    bgw.DoWork +=new DoWorkEventHandler(DoWork);  
    bgw.RunWorkerCompleted +=new RunWorkerCompletedEventHandler(RunWorkerCompleted);  
    bgw.RunWorkerAsync();  
}  
  
static void DoWork(object sender, DoWorkEventArgs e)  
{  
    //...  
}  
  
static void RunWorkerCompleted(object sender, RunWorkerCompletedEventArgs e)  
{  
    //...  
}
```

Task Parallel Library



```
static void TplStyle()  
{  
    Parallel.Invoke(  
        () => A(),  
        () => B(),  
        () => C(10)  
    );  
}
```

```
private static void A(){ /*...*/}  
private static void B(){ /*...*/}  
private static void C(int i){ /*...*/}
```

Task Parallel Library



```
for (int i = 0; i < values.Length; i++)  
{  
    values[i] = FastOperation(i);  
}
```

A screenshot of a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. The window contains the following text:

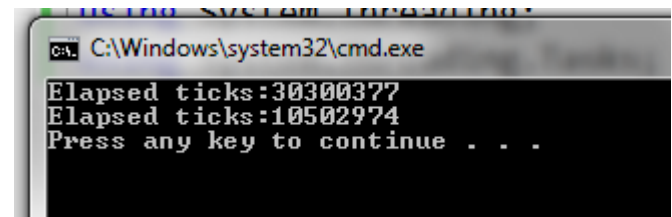
```
Elapsed ticks:164  
Elapsed ticks:3244  
Press any key to continue . . .
```

```
Parallel.For(0, values.Length, i =>  
{  
    values[i] = FastOperation(i);  
});
```

Task Parallel Library

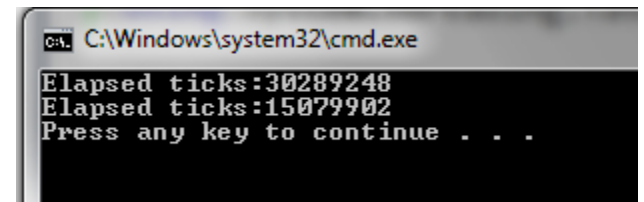


```
for (int i = 0; i < values.Length; i++)  
{  
    values[i] = SlowOperation(i);  
}
```



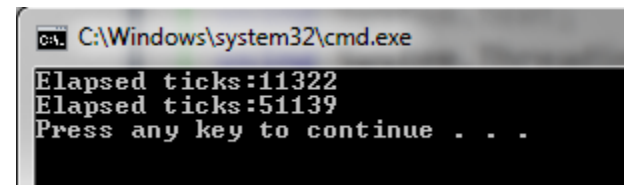
```
Parallel.For(0, values.Length, i =>  
{  
    values[i] = SlowOperation(i);  
});
```

```
var s = from v in values
        where SlowOperation(v) > (values.Length / 2)
        select v;
```

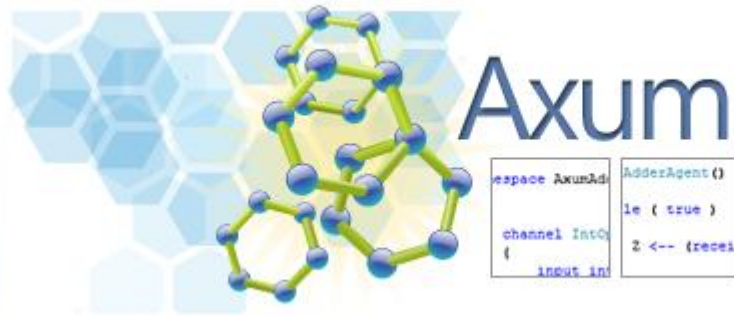


```
C:\Windows\system32\cmd.exe
Elapsed ticks:30289248
Elapsed ticks:15079902
Press any key to continue . . .
```

```
var p = from v in values.AsParallel()
        where SlowOperation(v) > (values.Length / 2)
        select v;
```



```
C:\Windows\system32\cmd.exe
Elapsed ticks:11322
Elapsed ticks:51139
Press any key to continue . . .
```



Axum

```
namespace Axum {
    AdderAgent()
    {
        channel IntC
        {
            input in
        }
    }

    AdderAgent()
    {
        le ( true )
        {
            2 <-- (receiv
        }
    }

    private writ
    {
        public %
        {
            var
        }
    }

    console.Writ
    {
        ar x = Int32.
        console.Writ
        {
            ar y = Int32.
            dder:X <-- x
            dder:Y <-- y
            console.Writ
    }
    }
}
```



atomic



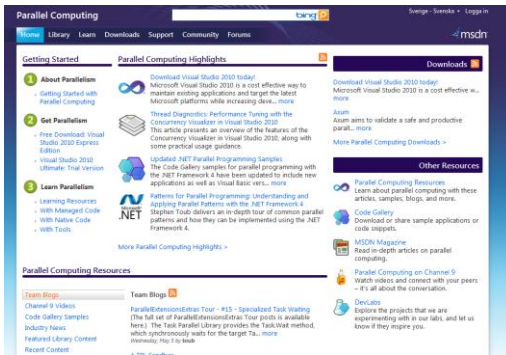
```
private void // Update s
{
    // Start a
    Atomic.Do(d
    {
        localVa
    }

    // Update s
    d++; // Upd
    eField(); //
    s(); // Cor
    d++; // Upd

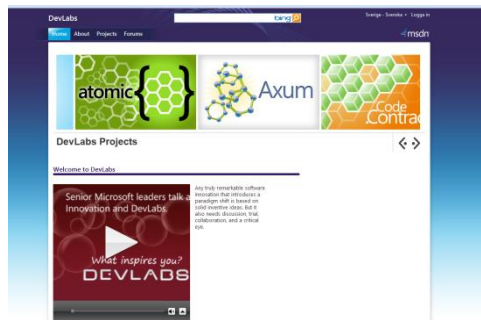
    AtomicDo()
    alVariable
    m_staticFi
    obj1 = new
}
```

Software Transactional Memory (STM.NET)

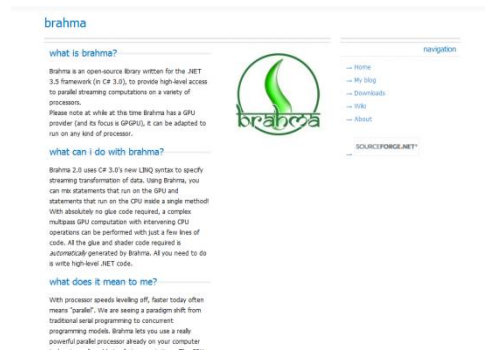
- msdn.microsoft.com/concurrency
- msdn.microsoft.com/devlabs
- brahma.ananthonline.net



The screenshot shows the MSDN website's 'Parallel Computing' section. It features a navigation bar with 'Home', 'Library', 'Learn', 'Downloads', 'Support', 'Community', and 'Forums'. The main content area is divided into several sections: 'Getting Started' with links for 'About Parallelism', 'Get Parallelism', and 'Learn Parallelism'; 'Parallel Computing Highlights' with articles on Visual Studio 2010 updates, performance tuning, and .NET Framework updates; 'Downloads' with links for Visual Studio 2010 Express Edition and Ultimate Trial Version; and 'Other Resources' including 'Parallel Computing Resources', 'Code Gallery', 'MSDN Magazine', and 'Parallel Computing on Channel 9'. A 'Team Blogs' section is also visible at the bottom.



The screenshot shows the MSDN 'DevLabs' page. It features a navigation bar with 'Home', 'About', 'Projects', and 'Forums'. The main content area includes a banner for 'atomic', 'Axum', and 'Code Contracts'. Below the banner is a 'DevLabs Projects' section with a 'Welcome to DevLabs' message. A featured article titled 'Senior Microsoft leaders talk innovation and DevLabs' is highlighted with a play button icon. The article text reads: 'My first remarkable software innovation that introduced a paradigm shift is based on this member blog. It is also worth discussion. I'd like to see a video on this.' The article is dated '11/11/2010'.



The screenshot shows the 'brahma' website. It features a navigation bar with 'Home', 'My blog', 'Downloads', 'Wiki', and 'About'. The main content area includes a 'what is brahma?' section with a description of the library and its purpose. A 'what can i do with brahma?' section lists various use cases, such as streaming transformation of data, parallel processing of images, and parallel processing of text. A 'what does it mean to me?' section discusses the benefits of parallel processing. The website also features a 'SOURCESFORGE.NET' logo and a 'navigation' menu.