

BIG>DAYS 2011



Parallel Data Processing

On-Premise und in der Cloud

Andreas Schabus

Academic Relations Manager
Microsoft Österreich GmbH
aschabus@microsoft.com

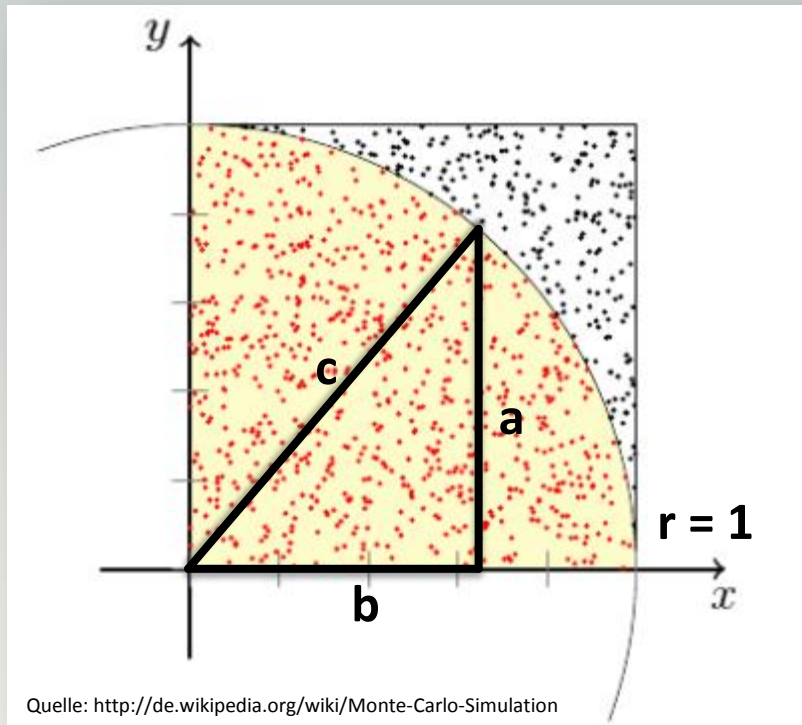
Rainer Stropek

Azure MVP, Developer, Entrepreneur
software architects
rainer@software-architects.at

Vorkenntnisse und Ziele

- Vorkenntnisse: C#
- Ziele:
 - Welche Gründe gibt es für parallele Programmierung?
 - Was bieten C# und .NET 4 dafür?
 - Welche Rolle spielt Parallelisierung in der Cloud?
 - Tipps, Tricks und typische Fallen
- Nicht-Ziele:
 - Training
 - High Performance Computing (HPC)





$$A = \frac{r^2 \pi}{4}$$

$$\sqrt{a^2 + b^2} \leq r$$

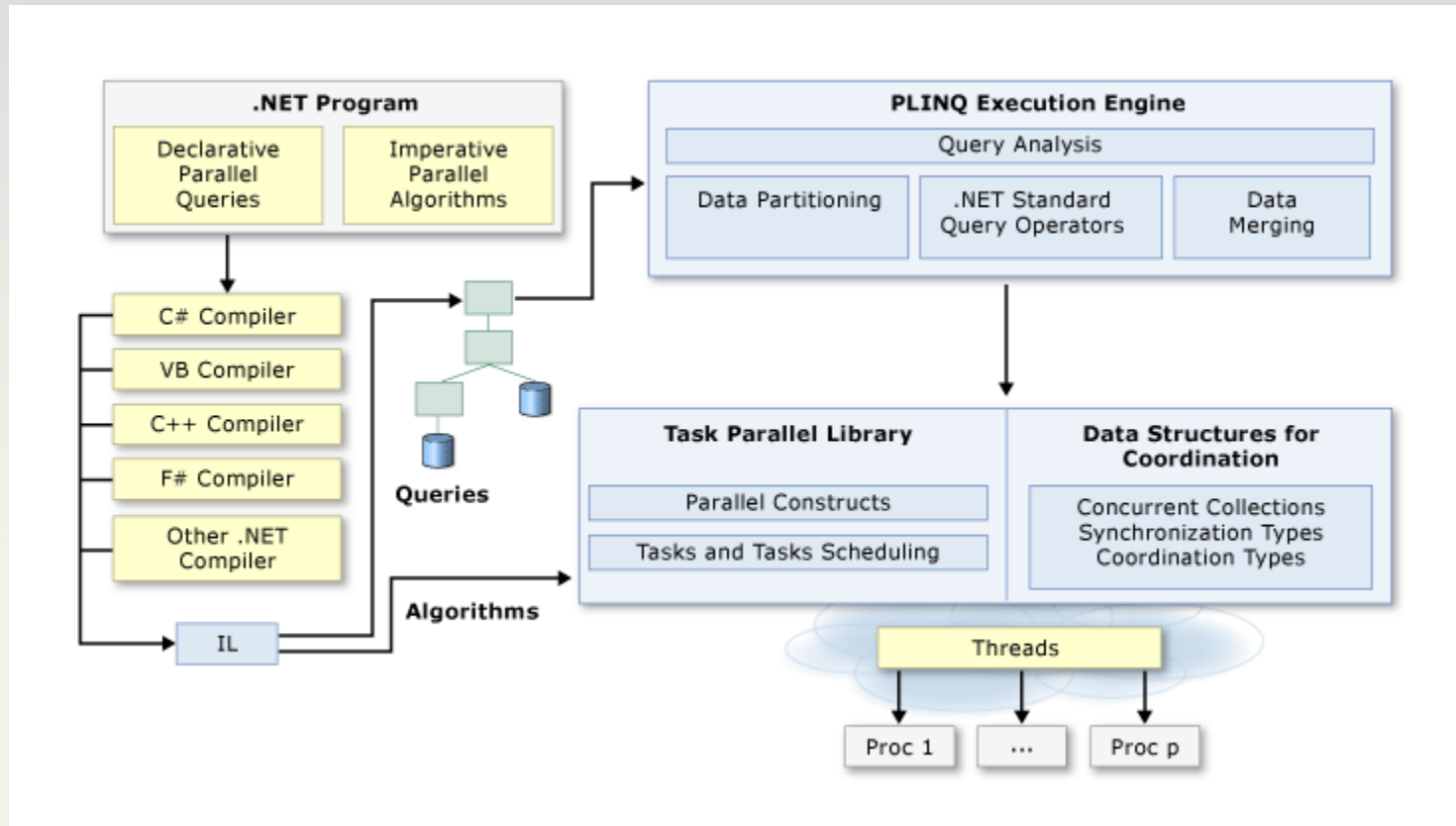
demo

Ermittlung von Pi mit Monte-Carlo-Simulation

Potenzial von paralleler Programmierung



Parallele Programmierung im .NET Framework



Siehe auch [Patterns of Parallel Programming](#)



Multithreading

Pre .NET 4

- [System.Threading](#) Namespace
- [Thread](#) Klasse
- [ThreadPool](#) Klasse

.NET 4

- [System.Threading.Tasks](#) Namespace
- [Task](#) und [Task<TResult>](#) Klassen
- [TaskFactory](#) Klasse
- [Parallel](#) Klasse



Kurzer Überblick über Tasks

- Starten
 - [Parallel.Invoke](#) (...)
 - [Task.Factory](#).StartNew (...)
- Warten
 - myTask.Wait ()
 - [Task](#).WaitAll
 - Task.WaitAny
 - Task.Factory.ContinueWhenAll (...)
 - Task.Factory.ContinueWhenAny (...)
- Verknüpfen
 - Task.Factory.StartNew (...,
[TaskCreationOptions](#).AttachedToParent) ;
- Abbrechen
 - Cancellation Tokens

Nicht in Silverlight ☹



demo

TPL und Tasks in C# und .NET 4

Live Coding



Schleifen – Parallel.For

```
for (int i = 0; i < 1000; i++)  
{  
    // Do Something  
}
```

```
Parallel.For(0, 1000, (i) =>  
{  
    // Do Something  
});
```



Schleifen – Parallel.For

```
var results = new bool[numberOfPointsPerLoop];  
Func<double, double, bool> testIfInside =  
    (x, y) => Math.Sqrt(x * x + y * y) <= 1;  
[...]
```

„Pure Funktion“

Paralleles Erzeugen des
Ergebnis-Arrays

Array → paralleler
Zugriff auf einzelne
Felder ist thread-safe

```
Parallel.For(0, numberOfPointsPerLoop,  
    i => results[i] =  
        testIfInside(  
            ThreadSafeRandom.NextDouble(),  
            ThreadSafeRandom.NextDouble()));
```



Schleifen – Parallel.ForEach

```
foreach (var item in GetMyList())  
{  
    // Do Something  
}
```

```
Parallel.ForEach(GetMyList(), item =>  
{  
    // Do Something  
});
```

Hoher Aufwand für abgesicherten
Zugriff auf MoveNext/Current
→ parallele Version oft langsamer



Was läuft hier falsch? (Code)

```
Console.WriteLine(  
    "Paralleles Lesen mit TaskFactory: {0}",  
    MeasuringTools.Measure(() =>  
        {  
            var tasks = new Task[urls.Length];  
            for (int i = 0; i < urls.Length; i++){  
                tasks[i] = Task.Factory.StartNew(() =>  
                    ReadUrl(urls[i]));  
            }  
  
            Task.WaitAll(tasks);  
        }  
    ));
```

Delegate verwendet Wert von i aus dem Main Thread →
IndexOutOfRangeException



Was läuft hier falsch? (Code)

```
// Variante 1
var tasks = new Task[url.Length];
for (int i = 0; i < url.Length; i++){
    var tmp = i;
    tasks[i] = Task.Factory.StartNew(() =>
        ReadUrl(url[tmp]));
}
```

Durch lokale Variable wird Delegate unabhängig;
mehr zum Thema unter dem Schlagwort *Closures*

```
// Variante 2
var tasks = new Task[url.Length];
for (int i = 0; i < url.Length; i++){
    tasks[i] = Task.Factory.StartNew(ReadUrl, url[i]);
}
```

State Object wird an Delegate übergeben



Von LINQ zu PLINQ

LINQ

```
var result = source
    .Where(...)
    .Select(...)
```

PLINQ

```
var result = source
    .AsParallel()
    .Where(...)
    .Select(...)
```

Aus IEnumerable wird
ParallelQuery

Tipp: AsOrdered() erhält die
Sortierreihenfolge



demo

PLINQ in C# und .NET 4

Live Coding

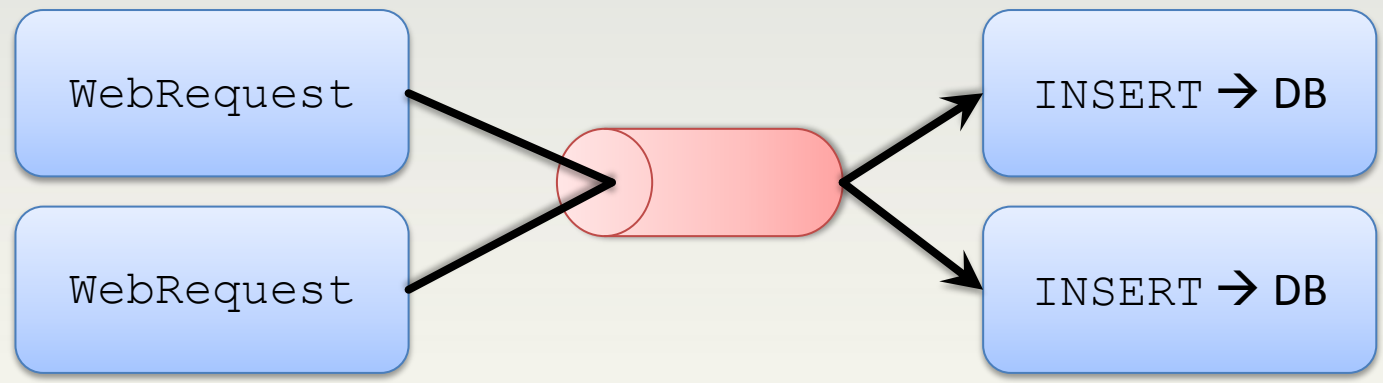
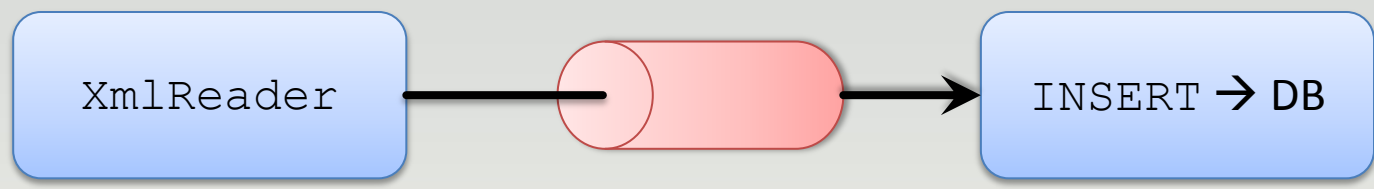


Performancetipps für PLINQ

- Allokieren von Speicher in parallelem Lambdaausdruck vermeiden
 - Sonst kann Speicher + GC zum Engpass werden
 - Wenn am Server: [Server GC](#)
- [False Sharing](#) vermeiden
- Bei zu kurzen Delegates ist Koordinationsaufwand für Parallelisierung oft höher als Performancegewinn
 - → Expensive Delegates
 - Generell: Auf richtige Granularität der Delegates achten
- `AsParallel()` kann an jeder Stelle im LINQ Query stehen
 - → Teilweise serielle, teilweise parallele Ausführung möglich
- Über `Environment.ProcessorCount` kann Anzahl an Kernen ermittelt werden
- **Messen, messen, messen!**



Producer/Consumer-Problem



demo

Producer/Consumer in C# und .NET 4

Live Coding



Collections für parallele Programmierung

- [System.Collections.Concurrent](#) für Thread-safe Collections
 - [BlockingCollection<T>](#)
Blocking- und Bounding-Funktionen
 - `ConcurrentDictionary<T>`
 - `ConcurrentQueue<T>`
 - `ConcurrentStack<T>`
 - `ConcurrentBag<T>`
- Optimal zur Umsetzung von Pipelines
 - Datei wird gelesen, gepackt, verschlüsselt, geschrieben



Parallele Programmierung On-premise vs. Cloud

- On-Premise
 - Multicore am Client- bzw. On-Premise-Server nutzen
 - Typischerweise Scale-up

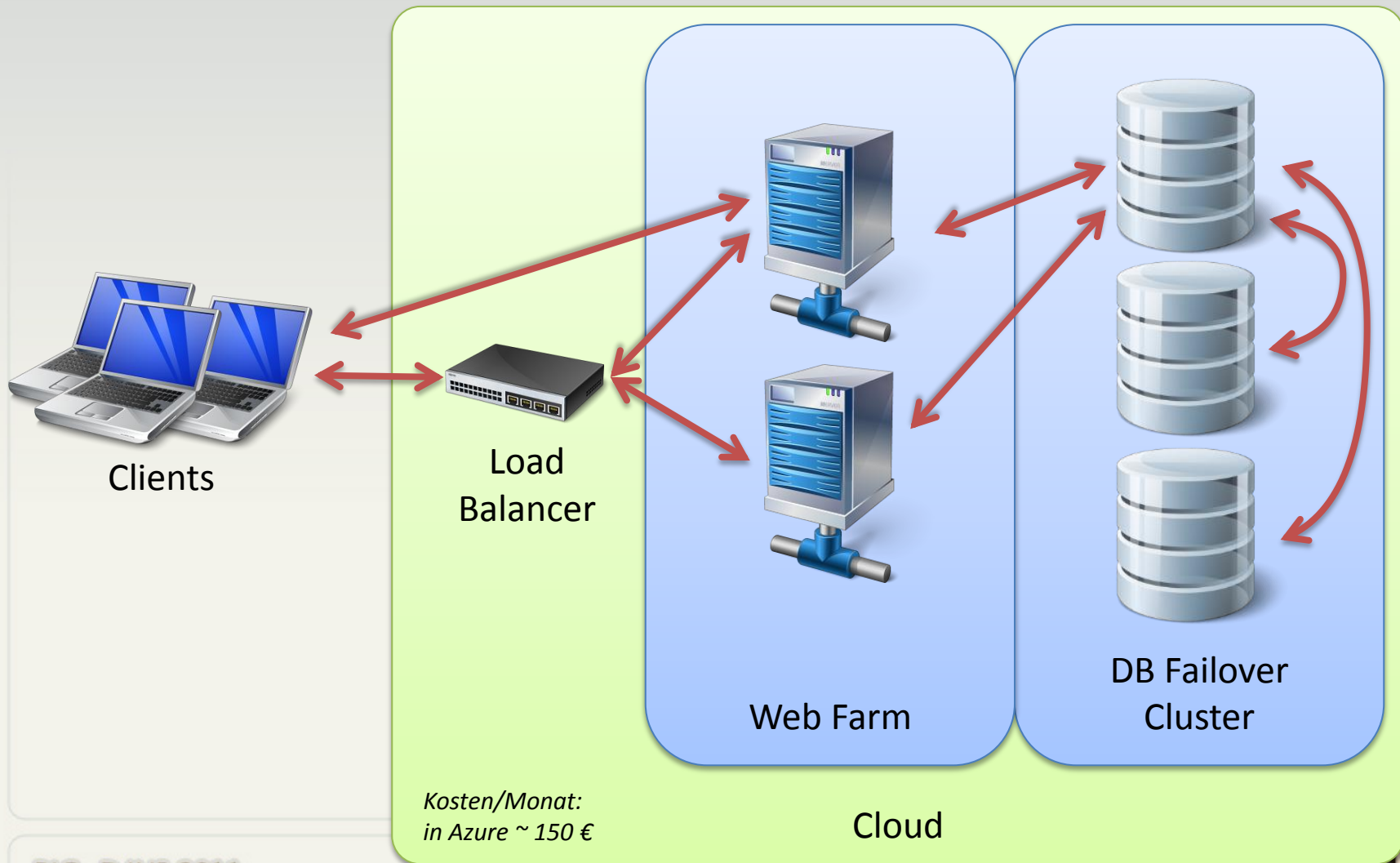


Parallele Programmierung On-Premise vs. Cloud

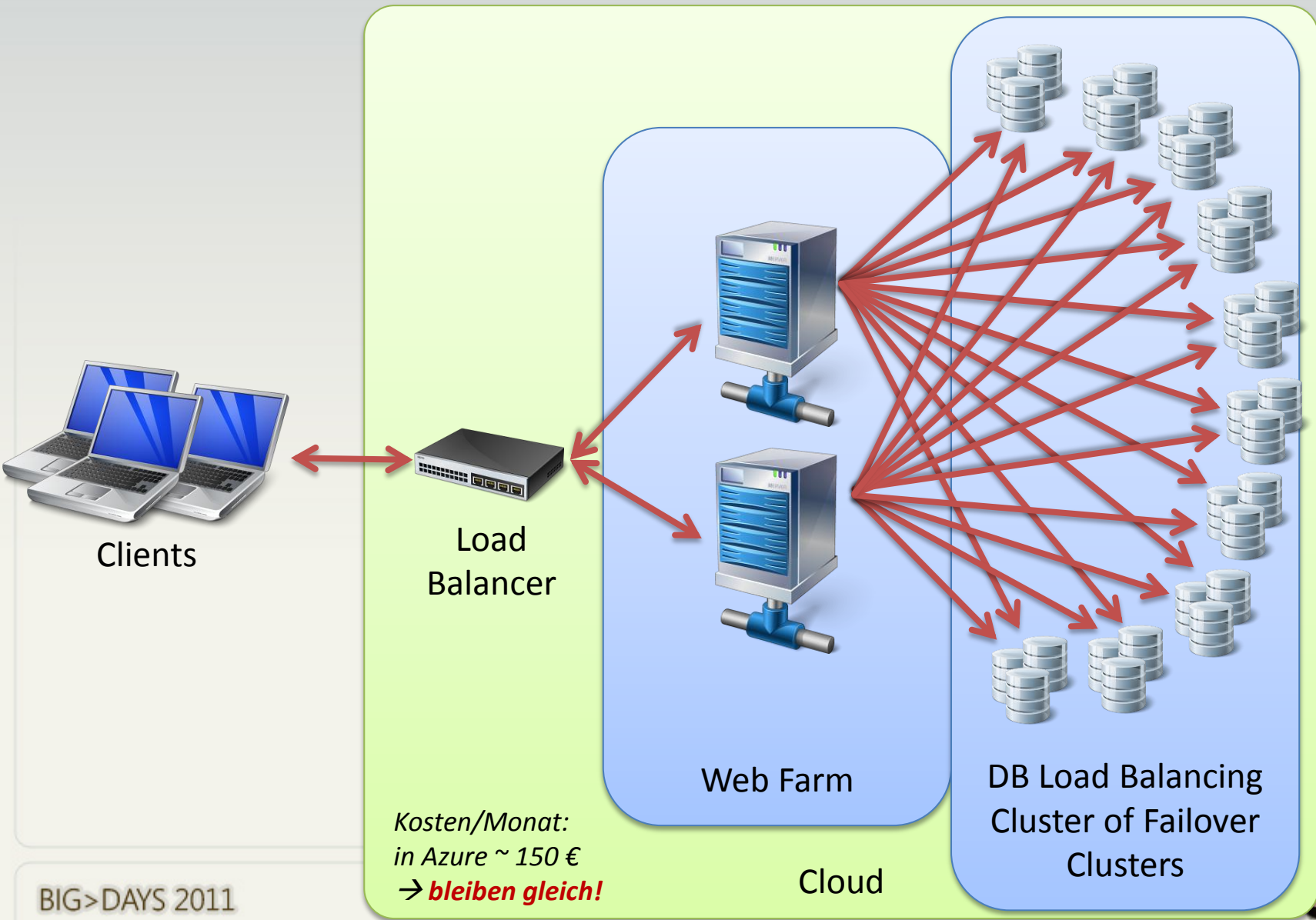
- Cloud
 - Scale-up: große Server zeitlich begrenzt
 - Kostenvorteil („Design to Cost“)
 - Wie vorher → Multicore muss genutzt werden!
 - Scale-out: Parallelisierung über Servergrenzen



Scale-up – nicht optimale Lösung



Sharding → Scale-out



Kosten/Monat:
 in Azure ~ 150 €
 → **bleiben gleich!**



Sharding LINQ Provider

```
[TestMethod]
public void ShardingTest()
{
    ThreadPool.SetMinThreads(10, 10);

    var result = CreateQueryableRoot()
        .Where(re => re.Location == "Wien"
            && re.HasBalcony.Value)
        .OrderBy(re => re.SizeOfGarden)
        .Take(25)
        .ToArray();
}
```

Versteckt Komplexität von
Sharding → identische
LINQ Queries



Ziele erreicht?

- Welche Gründe gibt es für parallele Programmierung?
 - Ermittlung von Pi
 - Optimale Ausnutzung der verfügbaren Ressourcen
- Was bieten C# und .NET 4 dafür?
 - TPL und Tasks
 - Parallel.For, Parallel.ForEach
 - PLINQ
 - System.Collections.Concurrent
- Welche Rolle spielt Parallelisierung in der Cloud?
 - Scale-up nur bedingt möglich
 - Kosten!
- Tipps, Tricks und typische Fallen



Weiterführende Literatur

- Toub, S.: [Patterns of Parallel Programming](#)
- MSDN: [Parallel Programming in the .NET Framework](#)
- Stropek, R.: [Customer OData Provider](#)
- Details zu „False Sharing“:
 - <http://msdn.microsoft.com/en-us/magazine/cc872851.aspx>
 - <http://software.intel.com/en-us/articles/reduce-false-sharing-in-net>



Give Feedback & Win!

Füllen Sie den Online Feedbackbogen aus und gewinnen Sie - Morgen in Ihrer Inbox!

Unter allen Teilnehmern verlosen wir:

- 2x LifeCam Studio Keyboard
- 2x Microsoft Arc Touch Mouse
- 2x WD Elements External Harddrive 2TB



	IT-PROFESSIONALS		DEVELOPERS		DYNAMICS
	The Future of IT-Professionals SAAL ALPHA	Real Life Szenarien <> „IT-Now“ SAAL BETA	Future of Development SAAL GAMMA	Modern Client Development SAAL DELTA	Business Solutions von Microsoft SAAL EPSILON
09:00 – 09:40	KEYNOTE				
09:50 – 11:00	Instant-on-Enterprise – the Future of IT-Pro	Weg von der IT als reine Infrastruktur hin zum strategischen Business-Werkzeug	Überblick und Architektur: Cloud-Lösungen mit Windows Azure planen	Webseiten mit ASP.NET MVC und Razor entwickeln	Integration von seiner besten Seite mit Microsoft Dynamics-Lösungen
11:20 – 12:30	Integration der Cloud in meine bestehende IT-Architektur Petra Korica-Pehserl	Wie man ein Deployment in den Sand setzt!	Azure Computing: die Worker-, Web-, und VM-Rolle	Visual Studio Light Switch – Das „Access“ der Zukunft mit Silverlight	Die neue Art des Verkaufens: Microsoft Dynamics CRM 2011
12:30 – 13:30	MITTAGSPAUSE				
13:30 – 14:40	Office 365 – Office, SharePoint, Exchange und Lync in der Cloud	Wie Sie mit Forefront Ihr Unternehmen absichern	Daten: strukturierter und unstrukturierter Speicher in der Cloud	Parallel Data Processing – On-Premise und in der Cloud	Der Microsoft Dynamics CRM Connector unter der Haube
15:20 – 16:30	Datenbanken und Business Intelligence in der Cloud	Lync Server 2010 – Unified Communications in der Praxis	Integration: On-Premise Services und Benutzerdaten an die Cloud anbinden	SharePoint Development für die Cloud mit Office 365	Mit Microsoft Dynamics CRM Online in die Cloud
16:50 – 18:00	Innovatives IT Management für den täglichen Gebrauch	SharePoint Large Volume Data Management	Migration: Bestehende Lösungen in die Cloud heben	From Scratch to Marketplace – Windows Phone 7 App-Entwicklung	Willkommen in der Zukunft: Was Sie mit Microsoft Dynamics AX 2012 erwartet

Mobile Website mit Session- & Speaker-Beschreibungen



Herzlichen Dank!

