

Erlang for .NET Developers

OJ Reeves

@TheColonial

<http://buffered.io/>



Welcome



Why am I doing this?



Why are you here?

- Because ...
 - ... you're a .NET hacker.
 - ... or you work with a .NET hacker.
 - ... or your boss paid for your ticket.

With that in mind ...

Why Erlang?

Perfect Use Case



September 2011 ...

```
$ netstat -an | grep -c EST  
1016313
```

That's **1,016,313** connections!

Perfect Use Case



January 2012 ...

```
jkb@c123$ sysctl kern.ipc.numopensockets  
kern.ipc.numopensockets: 2277845
```

That's **2,277,845** connections!

Perfect Use Case



January 2012 ...

```
CPU: 37.9% user, 0.0% nice  
13.6% system, 6.6% interrupt  
41.9% idle
```

```
Mem: 35G Active, 14G Inact, 18G Wired  
4K Cache, 9838M Buf, 27G Free
```


What's the goal?

- Highlight “sore” points in .NET.
- Ponder Erlang as an alternate.
- Dispel some FUD.
- Show that integration is easy.

Let's start with ...

Viriding's First Rule of Programming ...

“Any sufficiently complicated concurrent program in another language contains an ad-hoc, informally-specified, bug-ridden, slow implementation of half of Erlang.”

<http://rvirding.blogspot.com.au/2008/01/virdings-first-rule-of-programming.html>

Ask yourself ...

Why not **C#**?

Wrong question!

Ask yourself ...

Why not **F#**?

Wrong question!

Which question?

Why not **CCR**?

(Coordination & Concurrency Runtime)

What about ...

- Parallel Extensions?
- Task Parallel Library?
- 'async' keyword?
- ...

“Yet another Concurrency Attempt”

Key facets of Erlang

- Fault-tolerance is king.
- Concurrency is easier.
- Distribution is almost the same as concurrency.

What is it made of?

- Erlang – the language.
- OTP
 - Open Telecom Platform.
 - The libraries / Patterns & Practices.
- ERTS
 - Erlang RunTime System.
 - More of an OS than a VM.
- Even has its own built-in DB!

And guess what ...

It runs on Windows!



And guess what ...

It runs on Windows!

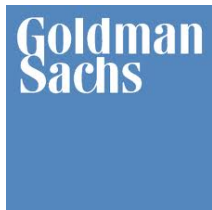
- Truly cross platform.
- Simple installer.
- R15 brings x64.

Erlang in a Tweet

Erlang: a battle-hardened, X-platform, functional lang that makes writing **reliable, concurrent, distributed** systems a joy.



But who uses it?



Good enough for them? Good enough for you!



Sweet Spot - Language

Pattern matching → awesome!

```
case { YourMove, TheirMove } of
  { rock, paper } -> lose;
  { paper, scissors } -> lose;
  { scissors, rock } -> lose;
  { _Move, _Move } -> draw;
  _ -> win
end.
```

Sweet Spot - Language

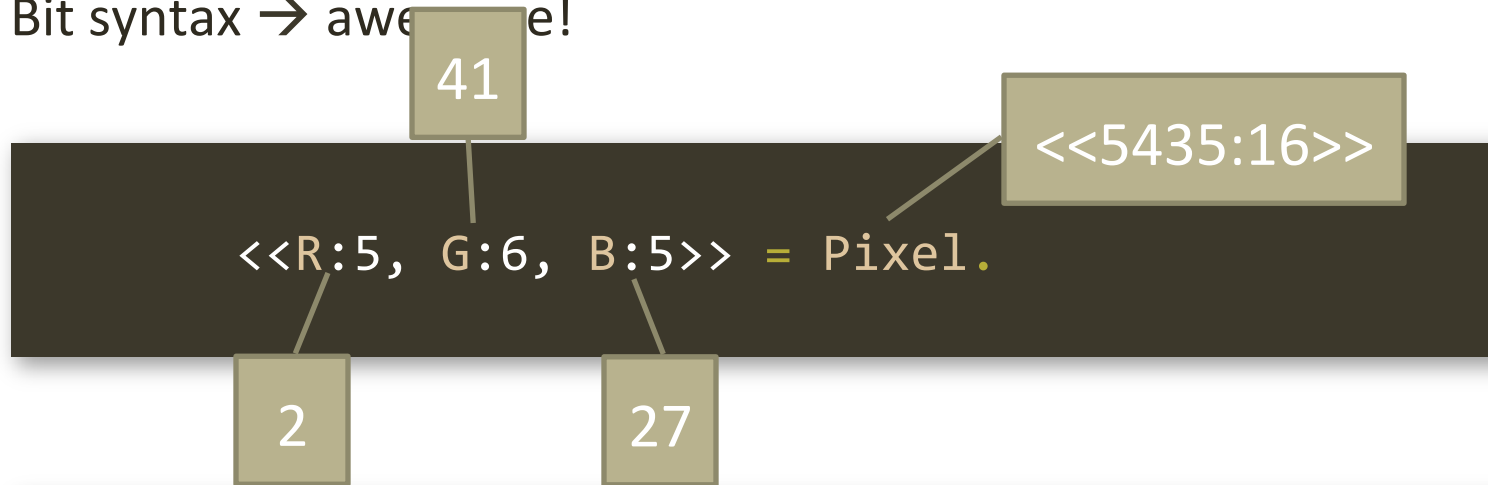
Bit syntax → awesome!

```
<<R:5, G:6, B:5>> = Pixel.
```

```
<<?IP_VERSION:4, HLen:4,  
  SvcType:8, TotLen:16, ID:16,  
  Flg:3, FragOff:13, TTL:8,  
  Proto:8, HdrChck:16, ScrIP:32,  
  DstIP:32, Rest/binary>> = Pkt.
```

Sweet Spot - Language

Bit syntax → awesome!



```
<<?IP_VERSION:4, HLen:4,  
  SvcType:8, TotLen:16, ID:16,  
  Flg:3, FragOff:13, TTL:8,  
  Proto:8, HdrChck:16, ScrIP:32,  
  DstIP:32, Rest/binary>> = Pkt.
```


Sweet Spot - Concurrency

```
printer() ->  
  Pid = spawn(simple_thread, print_msg, []),  
  Pid ! {msg, "Stop, you can't touch this"},  
  Pid ! stop.
```

```
print_msg() ->  
  receive  
    {msg, Msg} ->  
      io:format("~s~n", [Msg]),  
      print_msg();  
    stop ->  
      ok  
  end.
```

Concurrency .NET?

What is it like in .NET?



Concurrency .NET!

```
private string _msg;
private AutoResetEvent _msgSent = new AutoResetEvent(false);
private ManualResetEvent _finished = new ManualResetEvent(false);

public void Printer()
{
    var pid = new Thread(PrintMsg);
    pid.Start();
    _msg = "Stop, you can't touch this";
    _msgSent.Set();
    _finished.Set();
}

private void PrintMsg(object _)
{
    var waitHandles = new WaitHandle[] { _msgSent, _finished };

    while (true)
    {
        switch (WaitHandle.WaitAny(waitHandles))
        {
            case 0:
                Console.WriteLine("{0}", _msg);
                break;
            case 1:
                return;
        }
    }
}
```

Sweet Spot - Distribution

- Not an afterthought.
- Minimal code changes.
- So easy it's a sin.



What does it look like?

Sweet Spot - Distribution

```
printer() ->  
  Pid = spawn('foo@bar.com', simple_thread,  
    print_msg, []),  
  Pid ! {msg, "Stop, you can't touch this"},  
  Pid ! stop.
```

```
print_msg() ->  
  receive  
    {msg, Msg} ->  
      io:format("~s~n", [Msg]),  
      print_msg();  
    stop ->  
      ok  
  end.
```

Sweet Spot - Distribution

```
printer() ->  
  Pid = spawn('foo@bar.com', simple_thread,  
    print_msg, []).  
  Pid ! {msg, "Stop, you can't touch this"},  
  Pid ! stop.
```

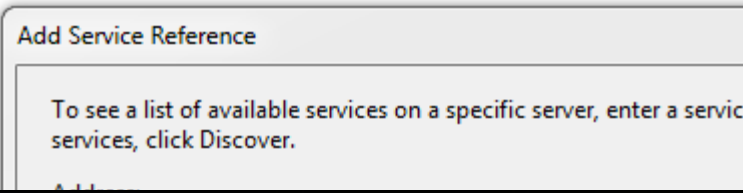
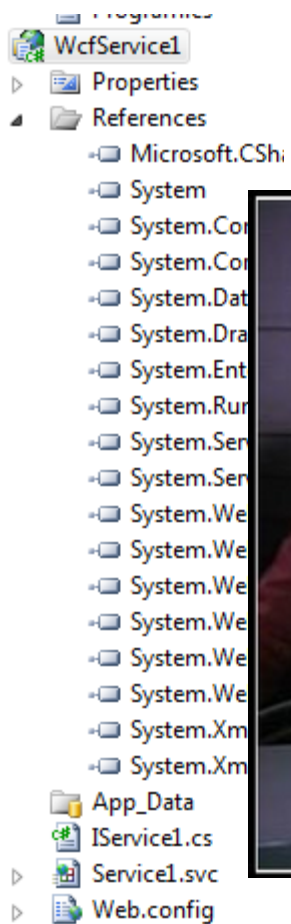
```
print_msg() ->  
  receive  
    {msg, Msg} ->  
      io:format("~s~n", [Msg]),  
      print_msg();  
    stop ->  
      ok  
  end.
```

Distribution .NET?

What is it like in .NET?



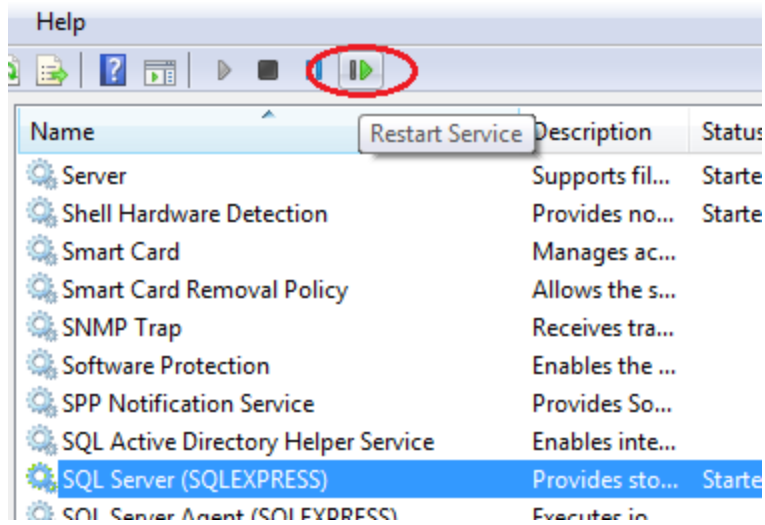
Distribution .NET!



```
Use the "Rename" command on the "Refactor" menu  
Service1 : IService1  
    setData(int value)  
    string.Format("You entered: {0}", value);  
  
DataContract(CompositeType  
    tion("composite"));  
  
    suffix";  
  
    value b  
    ing purp
```

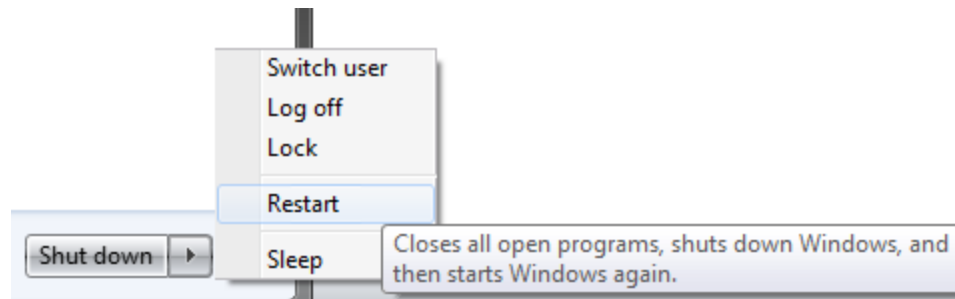
```
</serviceBehaviors>  
</behaviors>  
<serviceHostingEnvironment multipleSiteBindingsEnabled="true" />  
</system.serviceModel>  
<system.webServer>  
  <modules runAllManagedModulesForAllRequests="true"/>  
</system.webServer>
```

Availability .NET?



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.

C:\Users\OJ>iisreset
```



The system's down for maintenance AGAIN!?

?

- Server
- Shell Hardware Detection
- Smart Card
- Smart Card Removal Po
- SNMP Trap
- Software Protection
- SPP Notification Service
- SQL Active Directory He
- SQL Server (SQLEXPRES
- SQL Server Agent (SQL E



Sweet Spot - Availability

- Process linking and monitoring.
- Supervisor hierarchies.
- Hot code loading.
- Applications just don't stop.



.NET Baggage

Mutable state

Object orientation

Imperative approach

Lose your accent

Process	.NET (C#)	Erlang
Filter	<code>IEnumerable<T>.Where()</code>	<code>lists:filter()</code>
Map	<code>IEnumerable<T>.Select()</code>	<code>lists:map()</code>
Fold Left	<code>IEnumerable<T>.Aggregate()</code>	<code>lists:foldl()</code>
Concat Map	<code>IEnumerable<T>.SelectMany()</code>	<code>lists:flatmap()</code>
Create thread	<code>Thread.Start(),</code> <code>Thread.QueueUserWorkItem()</code>	<code>spawn(),</code> <code>spawn_link()</code>

Think more functionally!

Should you replace ...



Definitely not.

Should you replace ...



ASP.NET

Probably not.

... mmm perhaps!

Should you replace ...



For the sake of all that is good
in this world ...

YES!

Should you replace ...



Windows
Services

Yes!

Where Erlang shines

- Communications
- Message passing
- Work distribution
- Protocol handling
- High load, high connection count
- Mission critical systems

Where Erlang shines

If you don't want your systems to die

... use Erlang!

Integration

- No different to integrating .NET components
 - Web services
 - Comms via socket / TCP
 - Messaging (eg. RabbitMQ)
 - Client application drivers (eg. CorrugatedIron)

Integration

- A little different
 - ZeroMQ
 - NIF
 - Linked-in / port driver
- Serialisation / Deserialisation
 - Binary → Protocol buffers, MsgPack
 - Text → JSON, XML

Get amongst it

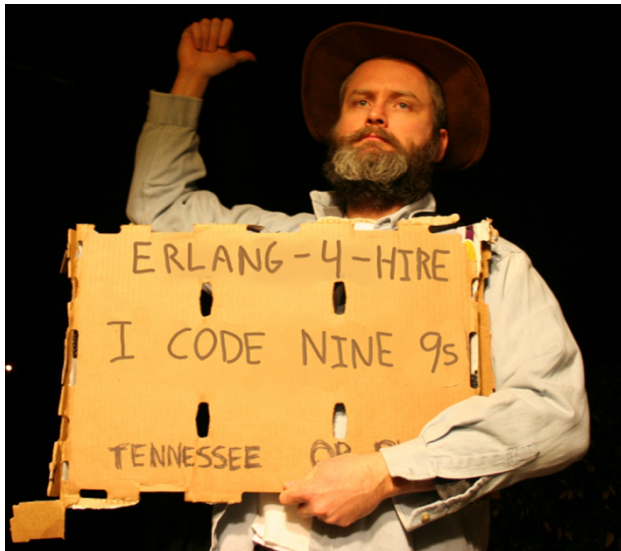
- <http://LearnYouSomeErlang.com/>
- Other books
 - “Erlang Programming” – Francesco C and Simon T
 - “Programming Erlang” – Joe A
 - “Erlang and OTP in Action” – Martin L, Eric M, & Richard C
- Kevin Smith’s screencasts

Other advice?

- Write code and lots of it.
- Contribute to OSS.
- Join the community – it's awesome.
- Talk to me!
 - @TheColonial
 - <http://buffered.io/>
 - oj@buffered.io

But wait ... there's more!

“Erlang for .NET Developers”



@bryan_hunter



@TheColonial

<http://groups.google.com/group/erl4net>

Other resources

- Podcasts
 - Talking Shop Down Under – OJ
 - Ep 55 – 17th May 2011
 - .NET Rocks – Bryan & OJ
 - Ep 753 – 27th March 2012
- Blogs
 - <http://codeswamp.com/>
 - <http://buffered.io/>

Thank you!



OJ Reeves
oj@buffered.io
@TheColonial
<http://buffered.io/>