



INTRO TO NODE.JS WITH MICROSOFT TOOLS

PRESENTED BY

JEFF AMMONS

MAY 2014

WHY NODE?

- Scalable
- Dynamic
- Common Language for front-end and back-end
- Fast turn around
- Case Studies
 - Walmart
 - PayPal
 - <http://www.zdnet.com/the-rewards-of-running-server-side-javascript-revealed-7000027060/>

WHY MICROSOFT TOOLS?

- Productivity
- Unified environment
- Node rising at Microsoft

WHY ME?

- My web development trajectory
 - HTML
 - Perl/CGI backend/HTML + JavaScript frontend
 - ASP backend/HTML + JavaScript frontend
 - Java backend/HTML + JavaScript frontend
 - ASP.Net Web Forms backend/HTML + JavaScript frontend
 - ASP.Net MVC backend/HTML/JavaScript/jQuery/jQueryUI/Knockout frontend
 - What's next?

I'M NOT SELLING NODE

- I'm beginning to use Node, but I'm not an evangelist
- Overall I still love ASP MVC
- JavaScript is required on the client, and is now an option on the server
- Bottom Line:
 - I don't know if you should use Node
 - I do know you had better at least be aware of it

VISUAL STUDIO

- Robust, easy debugging
- Intellisense
- Polished
- Doesn't inject too much crap
 - Mostly solution and project files

SQL SERVER

- Node doesn't *have* to be paired with NoSQL
- Node-sqlserver
 - <https://github.com/Azure/node-sqlserver>
 - Preview release
 - Other alternatives
 - I like mssql
 - <https://www.npmjs.org/package/mssql>

TYPESCRIPT

- Doesn't change JavaScript!
- Sits a layer above JavaScript
- Imposes *design time* structure
- Helps IDE catch errors and provide better Intellisense

AZURE

- Websites (Platform As A Service: PAAS)
 - Don't have to manage OS
 - Don't have to manage web server (directly)
 - Scalable
 - Git Deploy sites
 - iisnode
 - MSDN Credits - \$50 - \$150 per month
 - Standard Websites Acct: ~\$56/month
 - Bizspark

CAVEAT!!

- Azure Websites are AWESOME!
- Except when they aren't...

- The Blog-pocalypse
- Ghost blog engine
- Node based
- On Azure
- One day it just wasn't there



FALLOUT

- Switched to DocPad
 - Static HTML site generator
 - Also Node based
- May try again to figure out what went wrong
 - Some day...
- May switch to a VM

WHAT IS NODE?

- Javascript On The Server
 - *** Plus C or C++ at the core
 - Core *not* written in Javascript!
 - Google V8
 - ONE thread
 - Async or die!
 - Async is the default way to write
 - Never block thread or die!

THE PROBLEM WITH THREADS

- Creating threads takes time
- Each thread consumes 1 MB by default (in Windows)
 - Not so bad for one user on a desktop
 - Not so great for 100,000 users on a web server
- In ASP each user needs a thread
- They cannot connect until they can get a thread

WHAT ABOUT THREAD POOLING?

- Bad news
 - Saturation == death
 - Asynchronous Programming
helps but doesn't remove the
problem completely
 - Web farm helps
 - Default 300 – 350 concurrent
users

A screenshot of the MSDN Magazine website. The page has a light blue header with the 'MSDN Magazine' logo on the left and a search bar on the right. Below the header is a navigation menu with links: Home, Topics, Issues and Downloads (which is highlighted), Script Junkie, Subscribe, Submit an Article, and RSS. The main content area shows a breadcrumb trail: 'MSDN Magazine > Issues and Downloads > 2007 > March > WickedCode: Scalable Apps and Asynchronous Prog...'. The article title is 'Wicked Code' in a large, bold, dark blue font, followed by 'Scalable Apps with Asynchronous Programming in ASP.NET' in a slightly smaller, bold, dark blue font. The author's name, 'Jeff Prosise', is listed below the title. A link for 'Code download available at: WickedCode2007_03.exe (202 KB)' is provided. There is a section titled '+ Contents' with a plus icon. The article text begins with 'Do you want to know a secret? A deep, dark, dirty secret? One that, if revealed, would cause great angst in the ASP.NET community and prompt shouts of "Aha!" from the anti-Microsoft crowd?'. It then discusses the scalability of ASP.NET web sites, mentioning a 'self-imposed glass ceiling' and 'Server unavailable' errors. The final paragraph explains the underlying reason: the common language runtime (CLR) thread pool becoming saturated, leading to requests being rejected with a 'Heck no!' response.

MSDN Magazine

Search MSDN Magazine with Bing

bing

Home Topics **Issues and Downloads** Script Junkie Subscribe Submit an Article RSS

MSDN Magazine > Issues and Downloads > 2007 > March > WickedCode: Scalable Apps and Asynchronous Prog...

Wicked Code

Scalable Apps with Asynchronous Programming in ASP.NET

Jeff Prosise

Code download available at: [WickedCode2007_03.exe](#) (202 KB)

+ Contents

Do you want to know a secret? A deep, dark, dirty secret? One that, if revealed, would cause great angst in the ASP.NET community and prompt shouts of "Aha!" from the anti-Microsoft crowd?

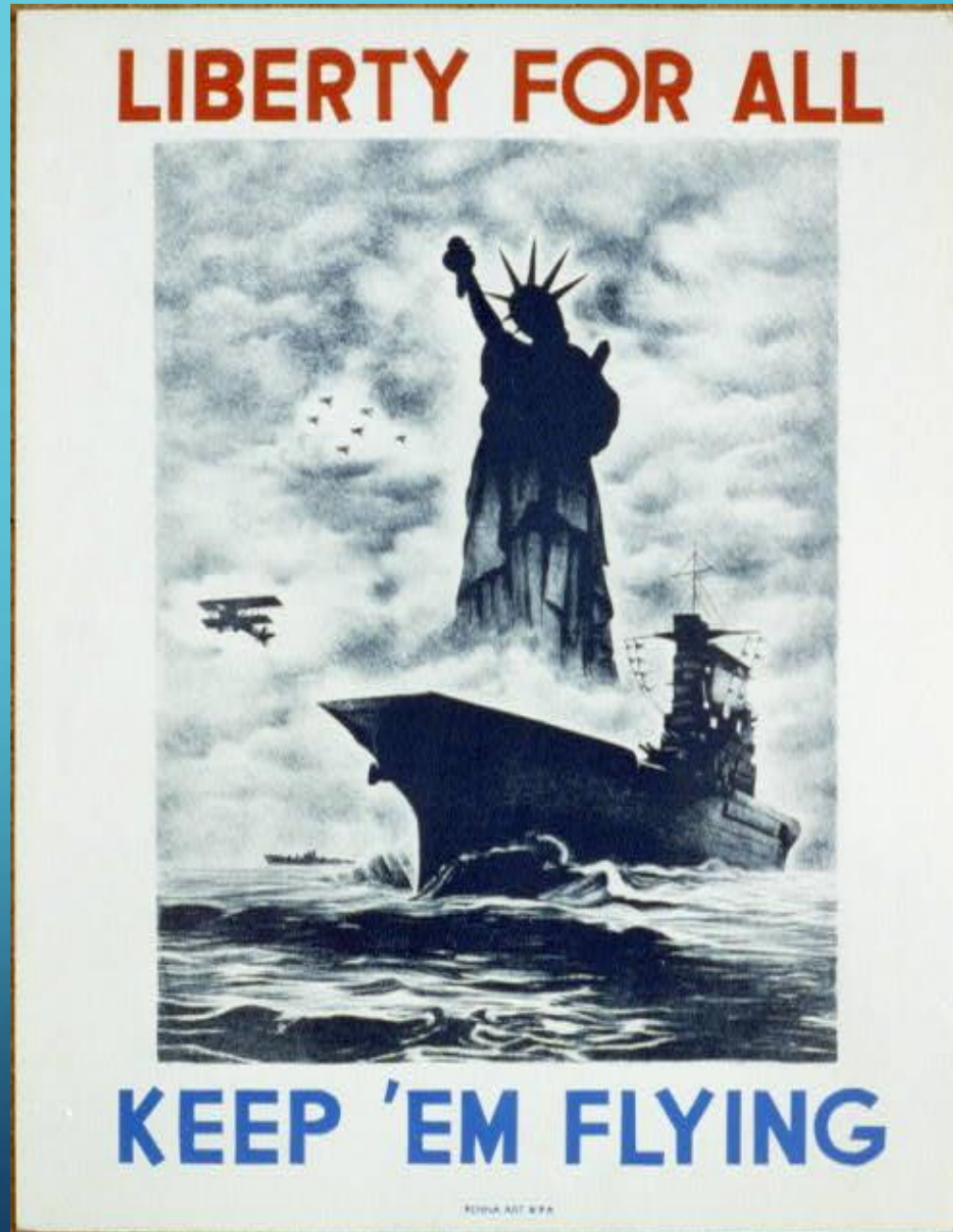
Most Web sites I've seen built with ASP.NET aren't very scalable, not because of a flaw in ASP.NET, but because of how the technology is used. They suffer a self-imposed glass ceiling that limits the number of requests they can process per second. These sites scale just fine until traffic rises to the level of this invisible ceiling. Then throughput begins to degrade. Soon after, requests start to fail, usually returning "Server unavailable" errors.

The underlying reason has been discussed many times in *MSDN® Magazine*. ASP.NET uses threads from a common language runtime (CLR) thread pool to process requests. As long as there are threads available in the thread pool, ASP.NET has no trouble dispatching incoming requests. But once the thread pool becomes saturated—that is, all the threads inside it are busy processing requests and no free threads remain—new requests have to wait for threads to become free. If the logjam becomes severe enough and the queue fills to capacity, ASP.NET throws up its hands and responds with a "Heck no!" to new requests.

<http://msdn.microsoft.com/en-us/magazine/cc163463.aspx>

ILLUSTRATION

- Fill pool with threads (planes)
- Wait for “passengers” (users)



ILLUSTRATION

- Each plane/thread picks up a passenger



ILLUSTRATION

- Everything's great until we have more passengers than planes...



ILLUSTRATION

- Everything's great until we have more passengers than planes...



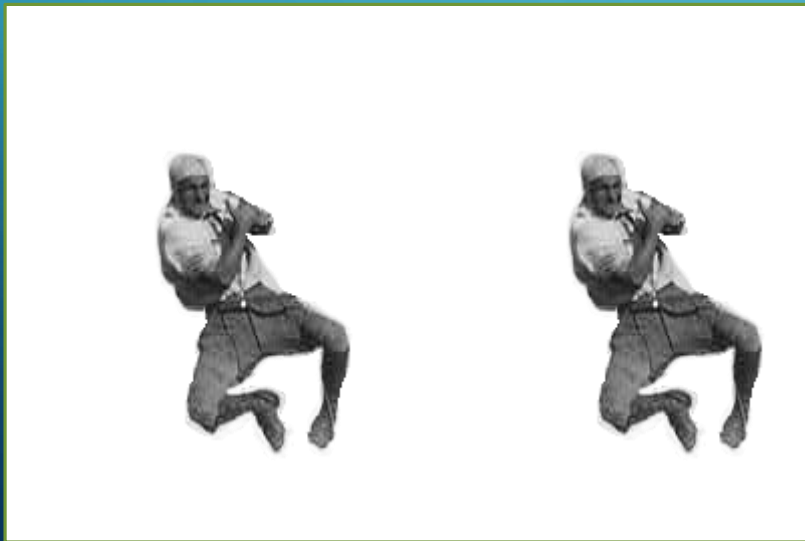
ILLUSTRATION

- Everything's great until we have more passengers than planes...



ILLUSTRATION

- Eventually passengers give up, planes give up



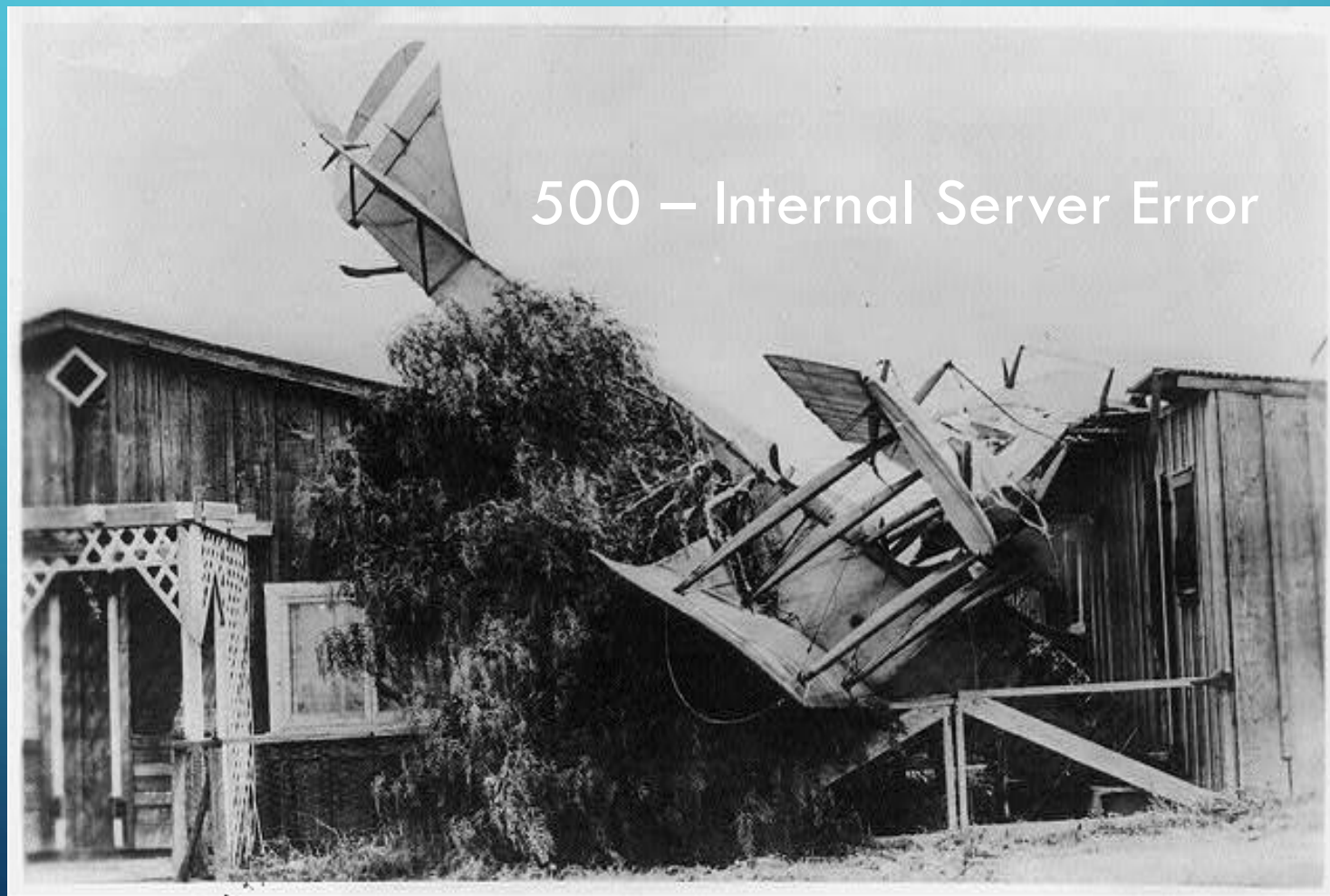
ILLUSTRATION

- Then this happens



ILLUSTRATION

- Then this happens



ASP IS IMPROVING

- Async and Await keywords
 - Help
 - Don't always work
- ASP **can** be asynchronous
 - But it's not the default
- Those planes crashed because no one expected that many passengers
- Node expects the passengers and expects YOU to expect them

BUT ONE THREAD? REALLY?

- One thread per INSTANCE
- iisnode
 - Create and load balance across multiple instances
 - <http://www.hanselman.com/blog/InstallingAndRunningNodejsApplicationsWithinIISOnWindowsAreYouMad.aspx>

GETTING STARTED

- Things You Must Install
 - Node.js
 - <http://nodejs.org/>
 - NPM (like Nuget)
 - Node.js Tools For Visual Studio
 - <https://nodejstools.codeplex.com/>

THINGS YOU MIGHT INSTALL

- Node Inspector
 - <https://github.com/node-inspector/node-inspector>
 - Debug in Chrome browser
- Azure SDK for Node.js
 - <http://www.azure.microsoft.com/en-us/develop/nodejs/>
- Node SQL Server
 - <https://github.com/Azure/node-sqlserver>
 - <https://www.npmjs.org/package/mssql>

DEMO 1

- Hello World

1. Via command line and text editor

1. `node hello.js`

2. Via Visual Studio

DEMO 2

- Debugging
 1. Via Node Inspector and Chrome
 1. `node-debug hello.js`
 2. Via Visual Studio

WHAT IS EXPRESS?

- Web framework
- Think of ASP MVC
 - Models
 - Views
 - Controllers
 - Routes

LAYERED FRAMEWORKS

- Connect (basic HTTP)
- Express (MVC)
- View Engines
- Other modules
 - LOTS of other modules

VIEW ENGINES

- Jade
 - Default view engine
 - Terseness a priority
 - jade-lang.com

jade

Node Template Engine

[Home](#)[Tutorial](#)[Language Reference](#)[API](#)[Command Line](#)[Online Demo](#)

```
doctype html
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript').
      if (foo) {
        bar(1 + 5)
      }
  body
    h1 Jade - node template engine
    #container.col
      if youAreUsingJade
        p You are amazing
      else
        p Get on it!
    p.
      Jade is a terse and simple
      templating language with a
      strong focus on performance
      and powerful features.
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Jade</title>
    <script type="text/javascript">
      if (foo) {
        bar(1 + 5)
      }
    </script>
  </head>
  <body>
    <h1>Jade - node template engine</h1>
    <div id="container" class="col">
      <p>You are amazing</p>
      <p>Jade is a terse and simple
      templating language with a
      strong focus on performance
      and powerful features.</p>
    </div>
  </body>
</html>
```

VASH

- Razor syntax engine
- @foo
- Like ASP MVC & Webpages
- <https://github.com/kirbysayshi/vash>

HOGAN

- Based on *Mustache*
- ‘Cause {{ looks like a *Mustache*, right?
- {{ blah }}
- From Twitter
- <http://twitter.github.io/hogan.js/>

HANDLEBARS

- Also based on Mustache
- Different flavor
- <https://www.npmjs.org/package/handlebars>

TONS OF OPTIONS

- Good and bad
- Good for flexibility
- Bad for standardization
- I'm not sure how you coordinate a team if everyone wants a different view engine

DEMO 3

- Simple Website with Express
 - Via Visual Studio
 - Command line is fine, but for the sake of time I'll just be doing Visual Studio from here on

DEMO 4

- Simple CRUD App
 - With SQL Server

MY LINKS

- jeffaBlog
 - <http://ammonsonline.com/>
- Twitter
 - @jeffa00
- Sci Fi, Comics And Goofiness
 - GalacticBeacon.com

ADDITIONAL (BETTER) RESOURCES

- Node.js For .Net Developers by Shawn Wildermuth
 - <http://pluralsight.com/training/Courses/TableOfContents/nodejs-dotnet-developers>
- ToDo App with Express.js/Node.js and MongoDB
 - <http://webapplog.com/todo-app-with-express-jsnode-js-and-mongodb/>
- Easy Node Authentication: Setup and Local
 - <http://scotch.io/tutorials/javascript/easy-node-authentication-setup-and-local>
- Jade Syntax Documentation By Example
 - <http://naltatis.github.io/jade-syntax-docs/>

ADDITIONAL ACKNOWLEDGEMENTS

- Images from Library Of Congress
 - Fearless Freddie
 - <http://www.loc.gov/pictures/item/2003655444/>
 - French Bi-plane crashed into house
 - <http://www.loc.gov/pictures/item/2003671113/>
 - Liberty For All
 - <http://www.loc.gov/pictures/item/96513610/>