

SharePoint Saturday
LOS ANGELES



SharePoint Saturday LOS ANGELES

Be sure to register and spread the word for two upcoming SharePoint Saturday Events

Ensenada

Saturday, October 12, 2019

CETYS University, Centro de Enseñanza Técnica y Superior, Ensenada, B.C 22860, Mexico

#SPSENS - <https://www.eventbrite.com/e/sharepoint-saturday-ensenada-2019-tickets-63037372478>

Denver

Saturday, October 12, 2019

Microsoft Technology Center - Denver, 7595 Technology Way #400, Denver, CO 80237

<https://www.eventbrite.com/e/sharepoint-saturday-denver-2019-tickets-63298384171?ref=estw>

SharePoint Saturday LOS ANGELES



We have two awesome User Groups here in Los Angeles check out their meetups!
LA SPUG: <https://www.meetup.com/Los-Angeles-SharePoint-User-Group-Meetup/>
LAExUG: <https://www.meetup.com/LAExUG/>

THANK YOU TO OUR SPONSORS

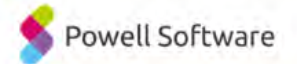
PLATINUM



GOLD



SILVER



BRONZE



PROVOKE

5

5A

LIGHTS, CAMERA, SHAREPINT!

SPSLA 2019 SharePint

Cinco

7241 W Manchester Ave.
Los Angeles, CA
90045
6:00pm – 8:00pm



Sponsored by:   PROVOKE

Code Solutions to Improve SharePoint Performance and Scalability via Caching

*@spmcDonough on Twitter
(for heckling purposes)*



Sean P. McDonough
Microsoft MVP
Bitstream Foundry LLC



Session Overview

Quick Introduction

Component Caching Options

- ASP.NET Cache
- Distributed Caching with Redis

Caching for Controls

- Fragment Caching
- Post-Cache Substitution

Q&A Throughout

okay ...

Why I



caching

Formerly the architect for a Fortune 25 company's publicly facing SharePoint presence

Highly trafficked environment with about 75,000 page views per hour (peak) in 2009

with about 75,000 page views
per hour (peak) in 2009

k) 1,000 requests/second into IIS



Supported initially with just 2 web
front ends (WFES). Eventually
moved to 4 WFES for growth.

... and finally

I got sick AND
tired of hearing
some people
complain that
"SharePoint
doesn't scale!!!"



In my experience ...



SharePoint
scaling and
performance
issues are not
the fault of the
platform itself.
I'm looking at
you, devs ...

platform itself.

I'm looking at
you, devs ...



What I'm about to
share might help

Let's get rolling



First up:

Component-Level Caching



Home

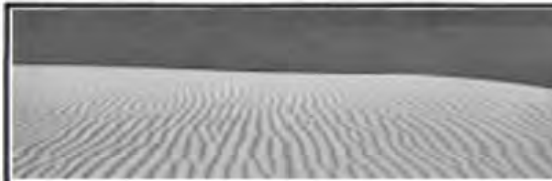
Dune Life

Giedi Prime Gazette

Mentat Monitor

Spice World

Welcome Back, Paul!



Today's Forecast

- Hot and Dry, 120F
- Slight Change of Worms

Right Now: **118F**



From Mentat News ...

- It is by will alone I set my mind in motion.
- It is by the juice of Sapho that thoughts acquire speed, the lips acquire stains, the stains become a warning.
- It is by will alone I set my mind in motion.

Spice Markets

(updated 15:23)



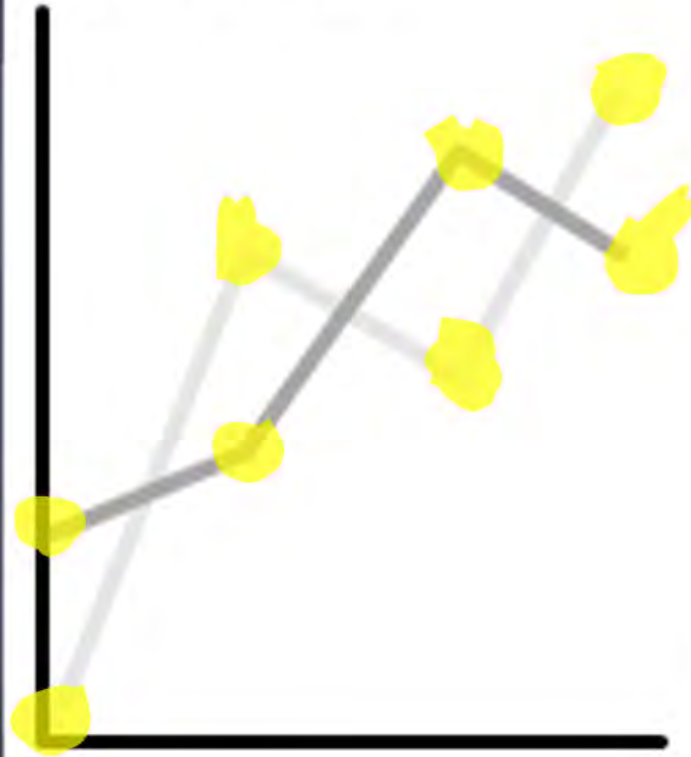


From
Mentat
News ...

It is by will alone I set
my mind in motion
It is by the juice of
Sopho that thoughts
acquire speed, the lips
acquire stains, the
stains become a
warning.
It is by will alone I set
my mind in motion

Spice Markets

(updated 15:23)



- Control rendering isn't complicated, but ...
- Data used is "expensive" (computation/latency)
- Need way to store expensive results between calls

- Control rendering isn't complicated, but ...
- Data used is "expensive" (computation/latency)
- Need way to store expensive results between calls

Two real options

ASP.NET
Cache

Distributed
Cache

ASP.NET Cache

public sealed class **Cache**
Member of [System.Web.Caching](#)

Summary:
Implements the cache for a Web application. This class cannot be inherited.

- System.Web.Caching.Cache class
- One instance per application domain
- Basically a key/value object dictionary
- In-memory use and thread-safe*
- Commonly accessed via Page and HttpContext objects
- Objects can be added with expiration windows, dependencies, & priority values
- Callbacks possible on object removal*

```

private String GetSomePi()
{
    // Attempt to retrieve a PI value from the ASP.NET Cache
    Object piValue = Cache[PI_VALUE_CACHE_KEY];

    // If the value isn't yet cached, compute it and cache it for later.
    if (piValue == null)
    {
        piValue = PiCalculator.Process(DIGITS_OF_PI_TO_COMPUTE);

        // Insert for indefinite time period
        Cache[PI_VALUE_CACHE_KEY] = piValue;

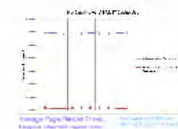
        //// Cache until a specific point in the future
        //Cache.Add(PI_VALUE_CACHE_KEY,
        //    piValue,
        //    null,
        //    DateTime.Now.AddSeconds(15),
        //    Cache.NoSlidingExpiration,
        //    CacheItemPriority.Normal,
        //    null);

        //// Cache for a sliding window of 3 seconds
        //Cache.Add(PI_VALUE_CACHE_KEY,
        //    piValue,
        //    null,
        //    Cache.NoAbsoluteExpiration,
        //    TimeSpan.FromSeconds(3),
        //    CacheItemPriority.Normal,
        //    null);
    }

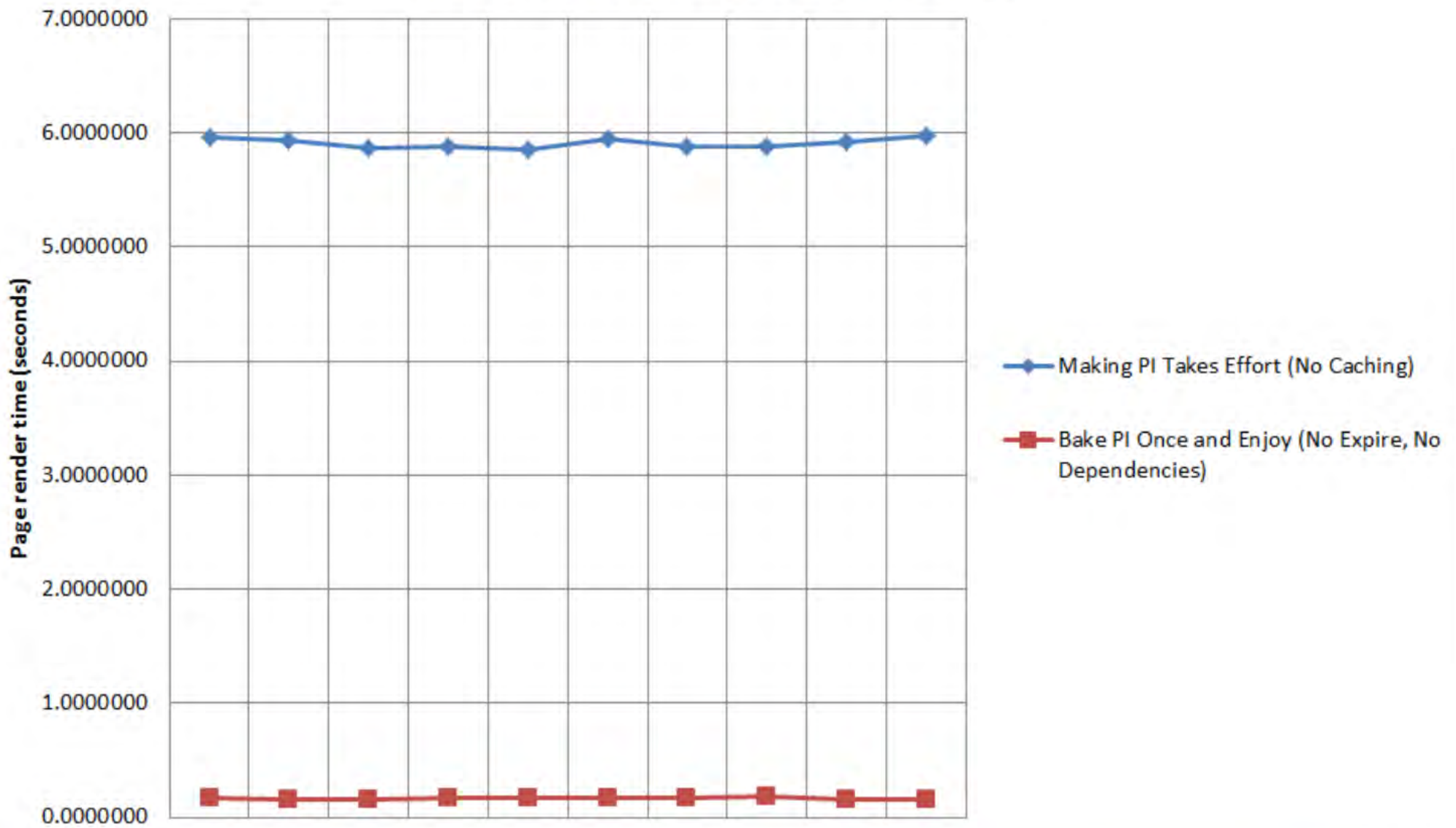
    return piValue.ToString();
}

```

Demo



No Caching vs. ASP.NET Cache Use



Average Page Render Times

Anonymous client-side request times;
10 samples each obtained using Fiddler

- No Caching: 5.909 sec
- ASP.NET Cache: 0.1641 sec

Limitations and Watch-Outs

- Not a durable store
- Don't assume something you put in will always be available
- Cache contents not available across WFEs in a load-balanced environment

Sum-up: Safe for general use. Just remember the cache is shared.

- Control rendering isn't complicated, but ...
- Data used is "expensive" (computation/latency)
- Need way to store expensive results between calls

Two real options

ASP.NET
Cache

Distributed
Cache

Distributed Cache

- Microsoft AppFabric (1.1) for Windows Server
- Provides highly available distributed caching
- Exists independent of SharePoint (and thus will work with any version of SharePoint)
- From a code perspective, very similar to writing code for the ASP.NET (local) cache
- Requires significant external configuration and setup, so you'll want to become good friends with your SharePoint administrators

Redis

Distributed Cache

- Redis is an open source (free!), in-memory data structure store
- Can be used as a database, cache, message broker, and more
- Easy to set up (unlike AppFabric) and administer; good tooling available (also free)
Easy to write code for (numerous client libraries) and integrate with your solutions

1 reference

```
private String GetSomePi()
{
    String piValue;

    // Set up the Redis client for cache item retrieval (and possible insertion)
    using (RedisClient redisCache = new RedisClient(REDIS_SERVER))
    {

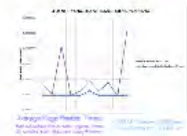
        // Attempt to retrieve a PI cache item from the Redis Cache
        Object piValueCacheItem = redisCache.Get<String>(PI_VALUE_CACHE_KEY);

        // If the value isn't yet cached, compute it and cache it for later.
        if (piValueCacheItem == null)
        {
            // The digits of Pi aren't yet computed, so do so now.
            piValue = PiCalculator.Process(DIGITS_OF_PI_TO_COMPUTE);

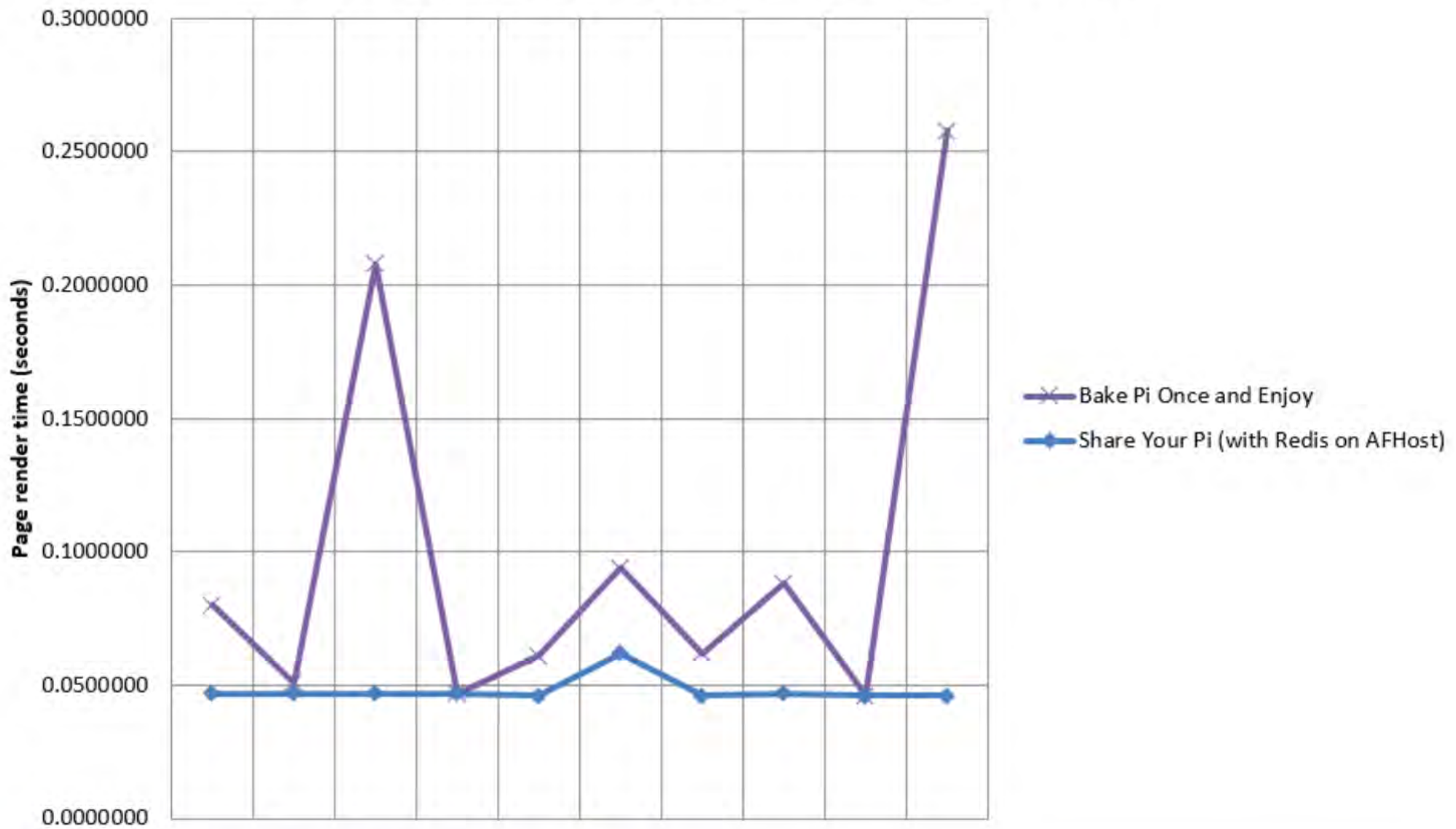
            // Insert the computed value of Pi into the Redis Cache
            redisCache.Set(PI_VALUE_CACHE_KEY, piValue);
        }
        else
        {
            // We got a non-null value back from the Redis cache, so simply convert
            // it to a String.
            piValue = piValueCacheItem.ToString();
        }
    }

    // That's it - return the String representing the computed value of Pi.
    return piValue;
}
```

Demo



ASP.NET Cache Use vs. Redis Distributed Cache



Average Page Render Times

Authenticated client-side request times;
10 samples each obtained using Fiddler

- ASP.NET Cache: 0.0995 sec
- Redis Distributed : 0.0481 sec

Limitations and Watch-Outs

- Redis **IS** a durable store, so whatever you cached will persist beyond the life of the current application/process
- Requires third-party libraries and installations.

Sum-up: Easy to setup and use - basically what we wished we had with Windows AppFabric Caching.

Next-up:

Control-Level
Caching





Home

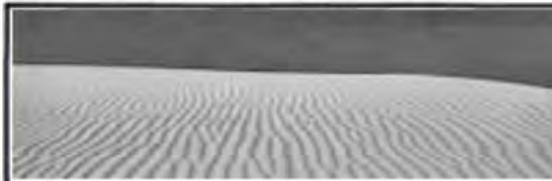
Dune Life

Giedi Prime Gazette

Mentat Monitor

Spice World

Welcome Back, Paul!



Today's Forecast

- Hot and Dry, 120F
- Slight Change of Worms

Right Now: **118F**

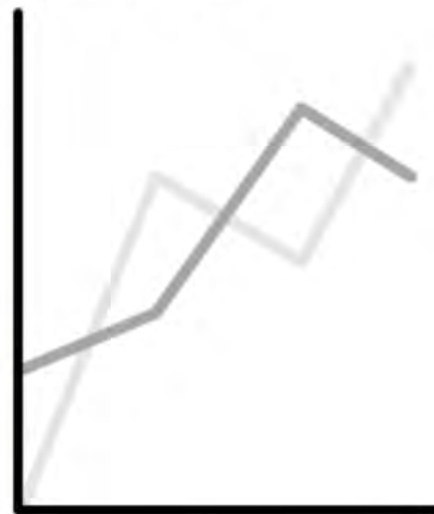


From Mentat News ...

- It is by will alone I set my mind in motion.
- It is by the juice of Sapho that thoughts acquire speed, the lips acquire stains, the stains become a warning.
- It is by will alone I set my mind in motion.

Spice Markets

(updated 15:23)





- Control displays static content or ...
- Entire HTML output block generated by control changes infrequently and/or according to predictable variables and patterns

Fragment Caching

An easily implemented way to cache the entire block of HTML that is generated by a control

To implement, simply add something like the following to an ASCX control file:

```
<%@ OutputCache Duration="120" VaryByParam="none" %>
```

(Common) options to vary output exist based on:

- HTTP Header
- Query string value (GET) or parameter (POST)
- Value of child control in ASCX

```
<%@ Assembly Name="$SharePoint.Project.AssemblyFullName$" %>
<%@ Assembly Name="Microsoft.Web.CommandUI, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce11
<%@ Register Tagprefix="SharePoint" Namespace="Microsoft.SharePoint.WebControls" Assembly="Microsoft.Sh
<%@ Register Tagprefix="Utilities" Namespace="Microsoft.SharePoint.Utilities" Assembly="Microsoft.Share
<%@ Register Tagprefix="asp" Namespace="System.Web.UI" Assembly="System.Web.Extensions, Version=3.5.0.6
<%@ Import Namespace="Microsoft.SharePoint" %>
<%@ Register Tagprefix="WebPartPages" Namespace="Microsoft.SharePoint.WebPartPages" Assembly="Microsoft
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="WeatherRightNowScrapper.ascx.cs" Inherits='
```

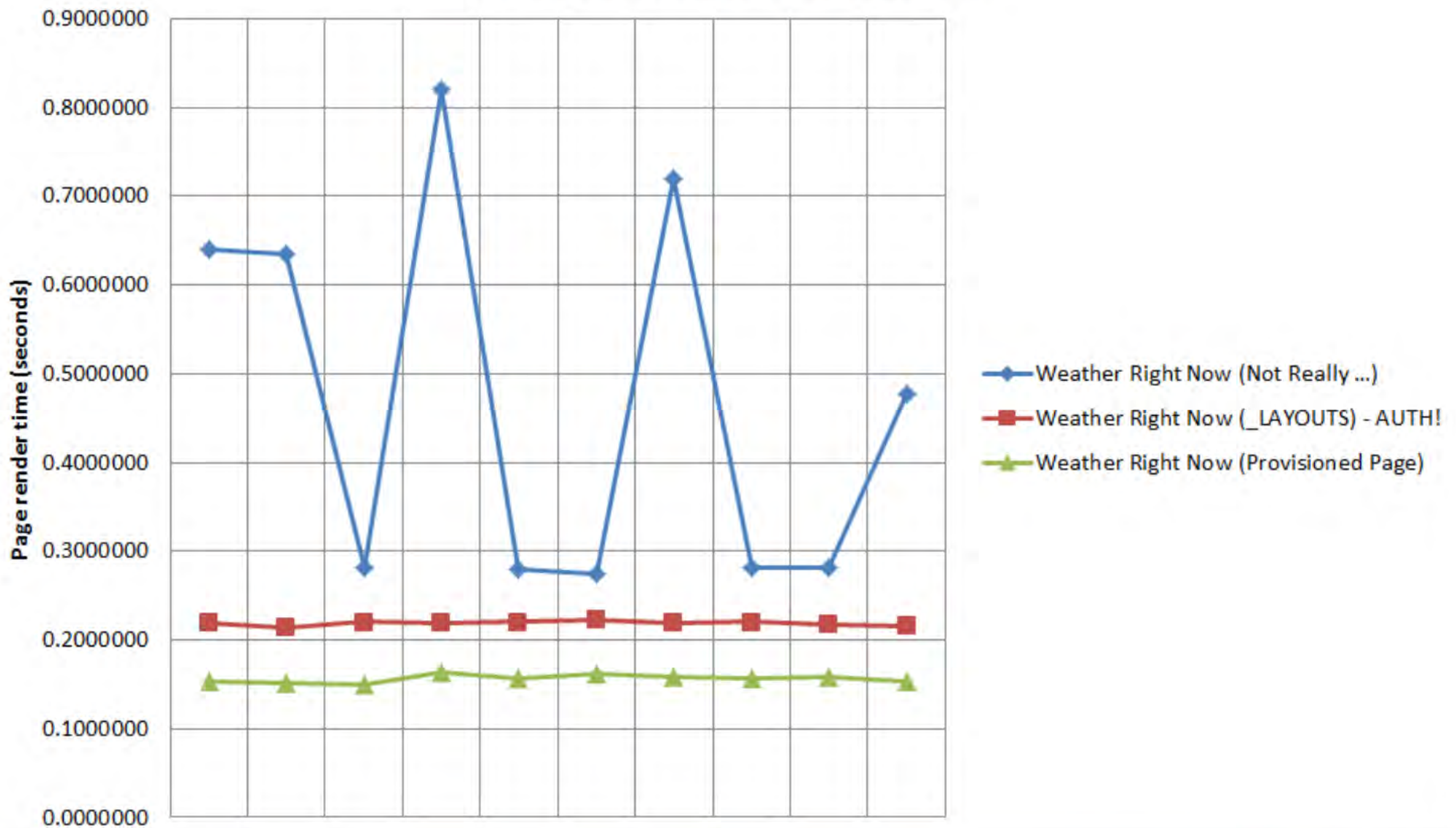
```
<%@ OutputCache Duration="120" VaryByControl="ZipCodeTextbox" %>
```

```
<asp:Panel ID="MainPanel" runat="server" BorderStyle="Solid" BorderWidth="1px" style="padding: 5px;">
    TIME WHEN CONTROL WAS RENDERED: <asp:Label ID="GenerationTimeLabel" runat="server"></asp:Label>
    <br />
    TIME TO GENERATE HTML OUPUT: <asp:Label ID="TimeToComputeLabel" runat="server"></asp:Label>
    <hr />
    ZIP CODE: <asp:TextBox ID="ZipCodeTextbox" runat="server">45244</asp:TextBox>
    <input id="SubmitButton" type="submit" value="Get Weather" />
    <br />
    <asp:Literal ID="WeatherLiteral" runat="server"></asp:Literal>
</asp:Panel>
```

Demo



Fragment Caching in Controls



Average Page Render Times

- No Caching (Safe Mode Parsing*): 0.4691 sec
- Fragment Caching (_LAYOUTS Page): 0.2187 sec
- Fragment Caching (Provisioned Page): 0.1566 sec

Limitations and Watch-Outs

- Test your VaryBy... parameter settings carefully
- If using both page-level and control-level caching, page-level will trump control-level (duration) settings
- If caching doesn't appear to work, consider that the safe mode parser may be engaged. Work around it with a provisioned page, _layouts page, or another (safe) alternative

Sum-up: Safe way to cache control content that changes infrequently



Home

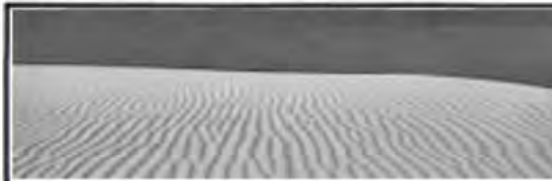
Dune Life

Giedi Prime Gazette

Mentat Monitor

Spice World

Welcome Back, Paul!



Today's Forecast

- Hot and Dry, 120F
- Slight Change of Worms

Right Now: **118F**



From Mentat News ...

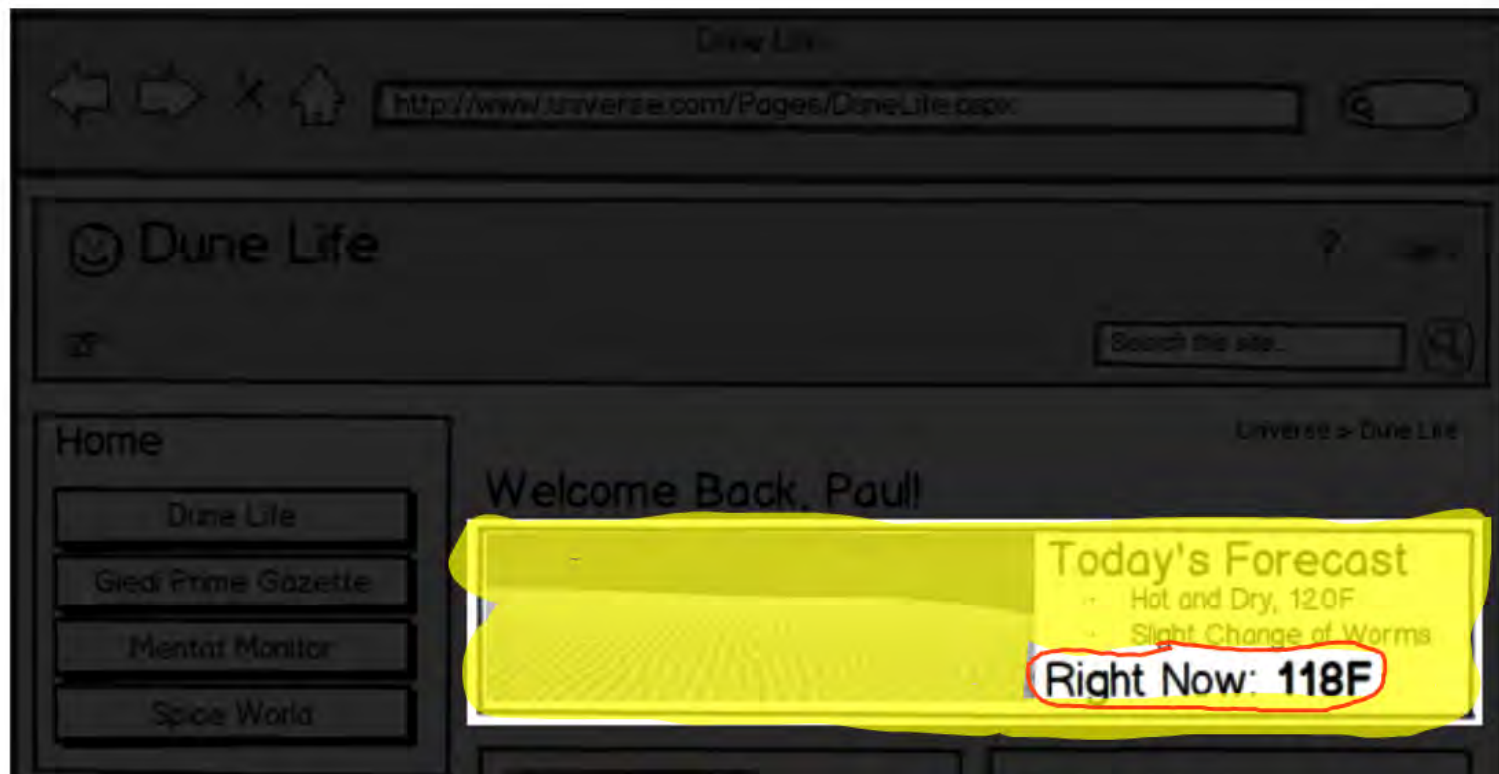
- It is by will alone I set my mind in motion.
- It is by the juice of Sapho that thoughts acquire speed, the lips acquire stains, the stains become a warning.
- It is by will alone I set my mind in motion.

Spice Markets

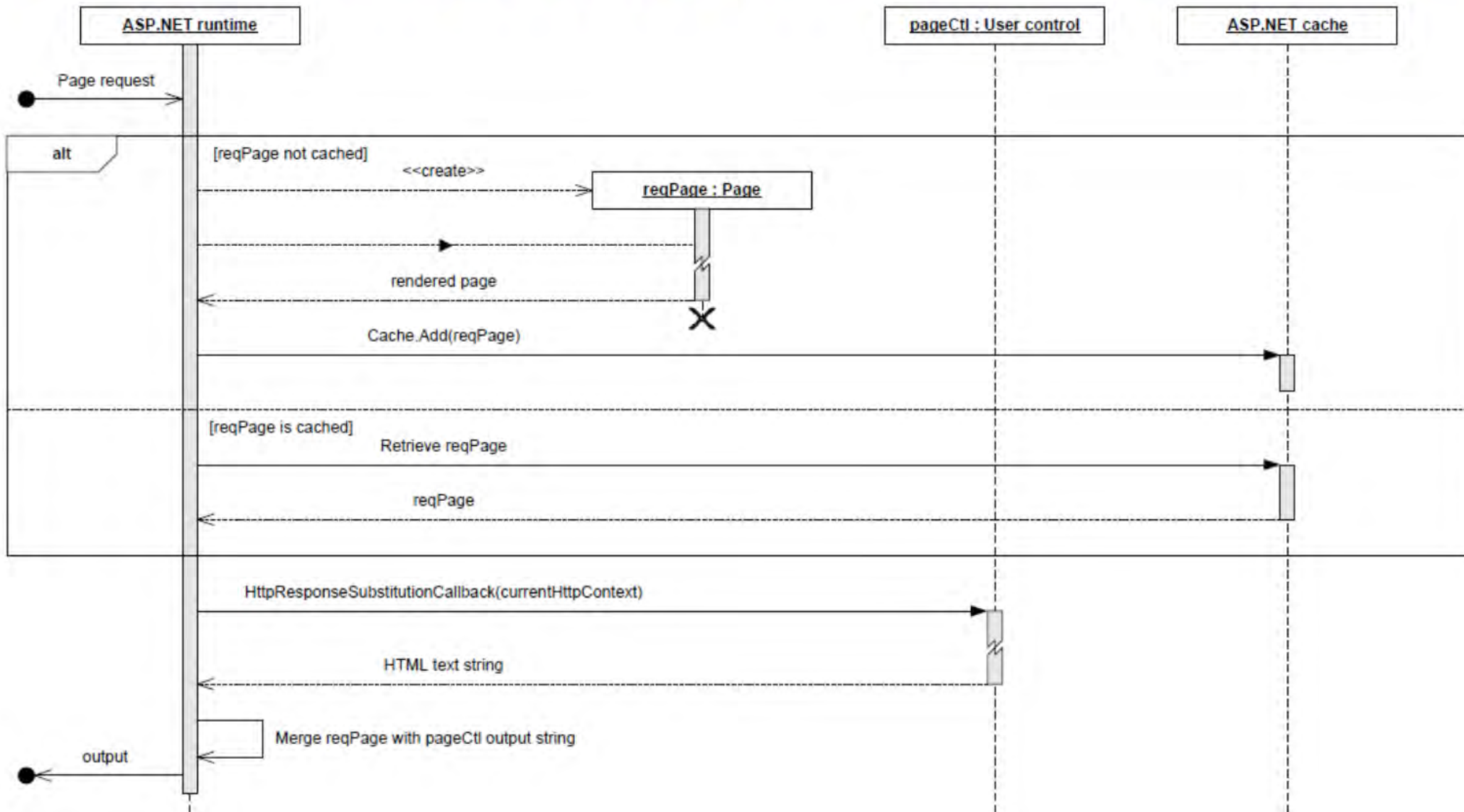
(updated 15:23)



- You are leveraging page output caching (i.e., the entire page's HTML output gets cached)
- Your control contains a mix of static and dynamic content
- You need a way to update the dynamic part (e.g., the "Right Now" temperature)



Post-Cache Substitution



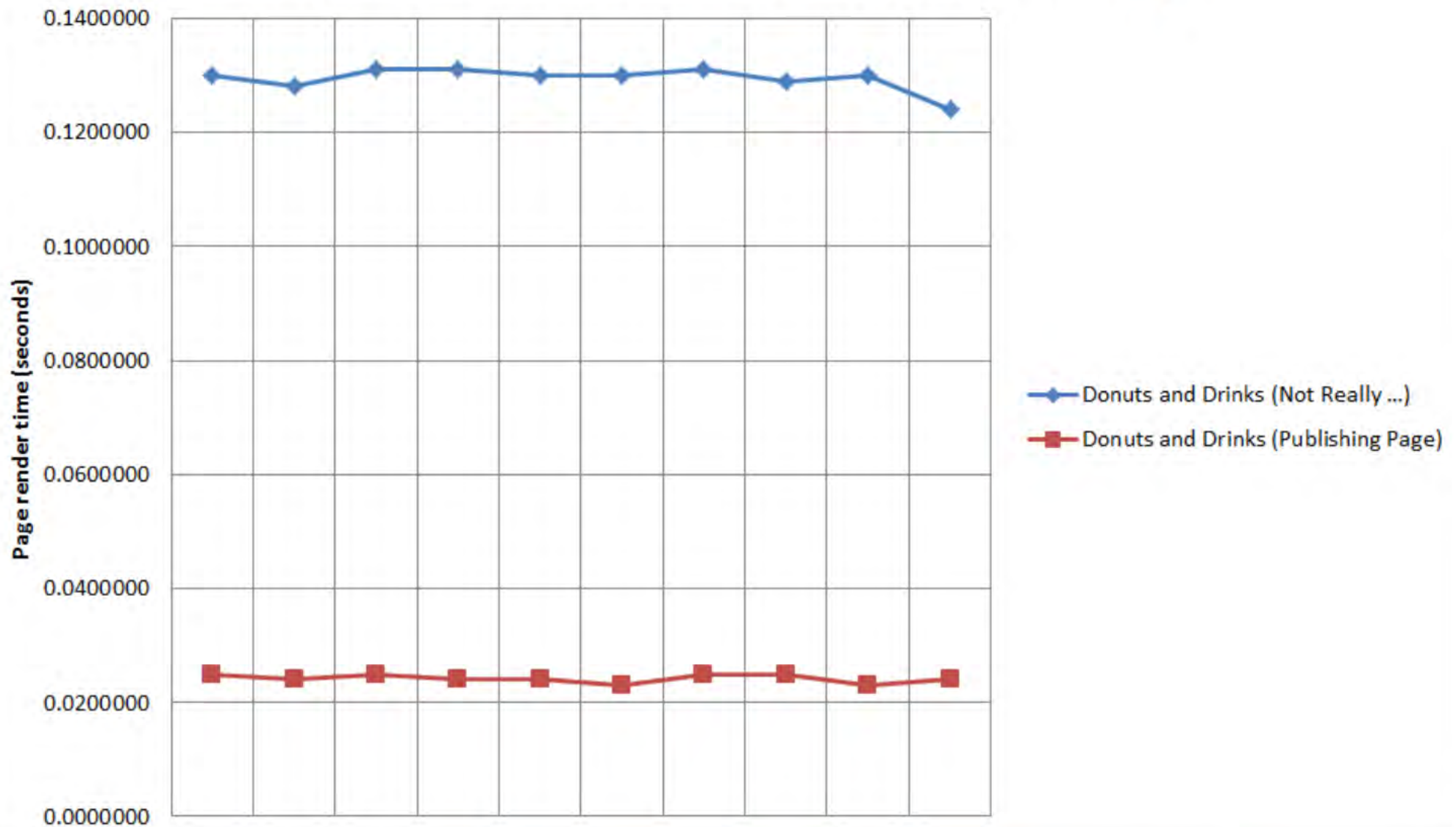
Page output caching "with benefits"



Demo

```
<asp:Panel ID="MainPanel" runat="server">
  <div style="text-align: center">
    <h2>Donut Caching: Now with Beverage!</h2>
  </div>
  <div style="position: relative;">
    <div style="float: left; width: 50%; text-align: center;">
      
      <table style="font-family: Arial, Helvetica, sans-serif; text-align: center; width: 100%">
        <tr>
          <td>
            Enjoy a tasty donut and ...
          </td>
        </tr>
        <tr>
          <td>
            Prepared at <asp:Label runat="server" ID="DonutPreparedLabel"></asp:Label>
          </td>
        </tr>
      </table>
    </div>
    <asp:Substitution runat="server" ID="BeverageSubstitution" MethodName="GetBeverageHtmlBlock" />
  </div>
</asp:Panel>
```

Post-Cache Substitution with Page Output Caching



Average Page
Render Times

- Page Output Cache Disabled: 0.1294 sec
 - Post-Cache Substitution (Pub Page): 0.0240 sec
- This is page output caching in action!*

Limitations and Watch-Outs

- Remember that SharePoint's page output caching is needed to actually make this work
- If caching isn't working at all, use the Debug Cache Information option to determine if the host page is being output cached
- Obscure issue: if you override rendering at the page level (e.g., within the master page), post-cache substitution will break

Sum-up: Great complement to page output caching for controls that contain some dynamic content

References

Cache Class (System.Web.Caching)

<http://msdn.microsoft.com/en-us/library/system.web.caching.cache.aspx>

Plan for feeds and the Distributed Cache service in SharePoint Server

<http://technet.microsoft.com/en-us/library/jj219572.aspx>

How To Perform Fragment Caching in ASP.NET by Using Visual C#.NET

<http://support.microsoft.com/kb/308378>

OutputCacheParameters Class

[http://msdn.microsoft.com/en-us/library/ms153449\(v=vs.90\)](http://msdn.microsoft.com/en-us/library/ms153449(v=vs.90))

Dynamically Updating Portions of a Cached Page

[http://msdn.microsoft.com/en-us/library/ms227429\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms227429(v=vs.90).aspx)

References

Substitution Class

[http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.substitution\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.substitution(v=vs.90).aspx)

Html Agility Pack

<https://html-agility-pack.net/>

Redis

<http://redis.io>

Redis on Windows

<https://github.com/MicrosoftArchive/redis/releases>

Redis Desktop Manager

<https://sourceforge.net/projects/redis-desktop-manager.mirror/>



Sean P. McDonough SharePoint Gearhead, Developer, AND Problem-Solver

Company:



www.bitstreamfoundry.com

Email: sean@bitstreamfoundry.com

Twitter: [@spmcdonough](https://twitter.com/spmcdonough)

Blog: <http://SharePointInterface.com>

About: <http://about.me/spmcdonough>