Visual Studios Profiling Tools

Timur Pratama Wiradarma

What is VS Profiling Tools?

- Integrated Profiling Tools in Visual Studio (Premium, Professional and Ultimate)
- Fast and Easy to implement
- Professional Look

What is VS Profiling Tools?

- CPU utilization
- I/O Bounds
- User Function
- .Net Memory Allocation
- Concurrency
- Tier Interaction
- Javascript

Target Metrics

- .NET-based applications
- Native Code C or C++

Technique

- Sampling
- Instrumentation
- .Net Memory Allocation
- Concurrency Visualizer

Sampling

- Initial Exploration of Application
- interrupts the computer processor at set intervals
- Collect function callstack
- Low Overhead Profiling (not instrusive)

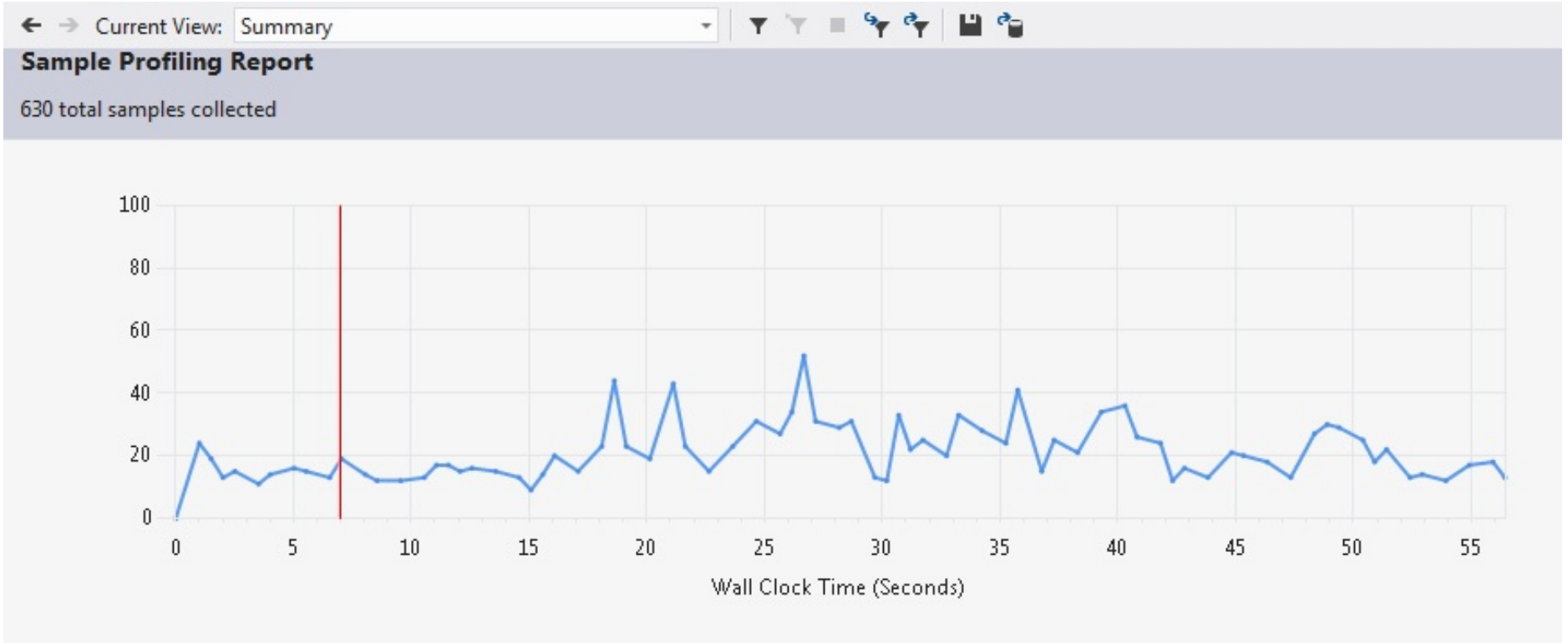
Sampling Event

- Clock Cycles (Default Interval 10,000,000)
- Page Faults
- System Calls
- Performance Counter

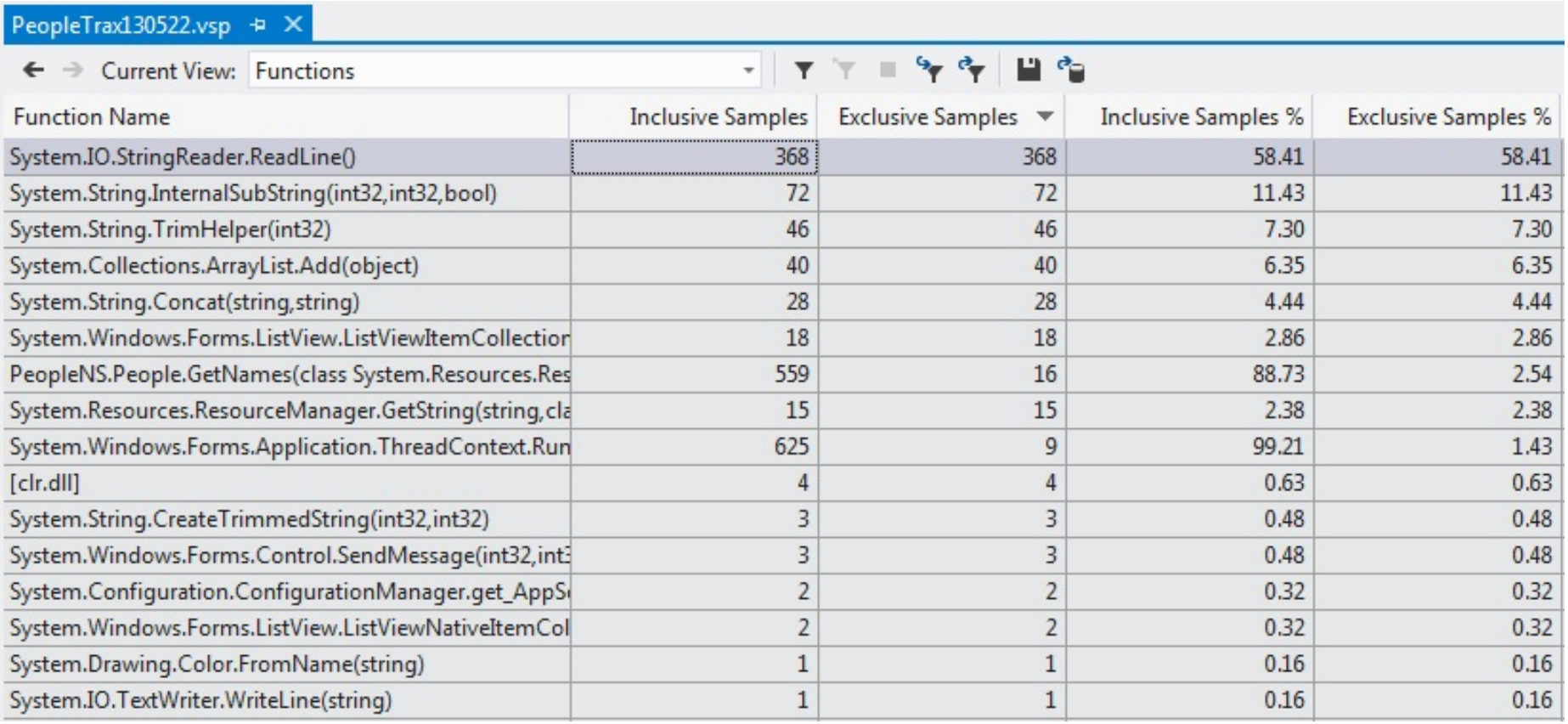
Sampling Representation

Instrumentation Function View Column Name	Definition
Inclusive Sample	The total number of samples that are collected during the execution of the target function.
Exclusive Sample	The number of samples that are collected during the direct execution of the instructions of the target function.
Inclusive Percent	The percentage of the total number of inclusive samples in the profiling run that are inclusive samples of the function or data range.
Exclusive Percent	The percentage of the total number of exclusive samples in the profiling run that are exclusive samples of the function or data range.

UI Representation



UI Representation

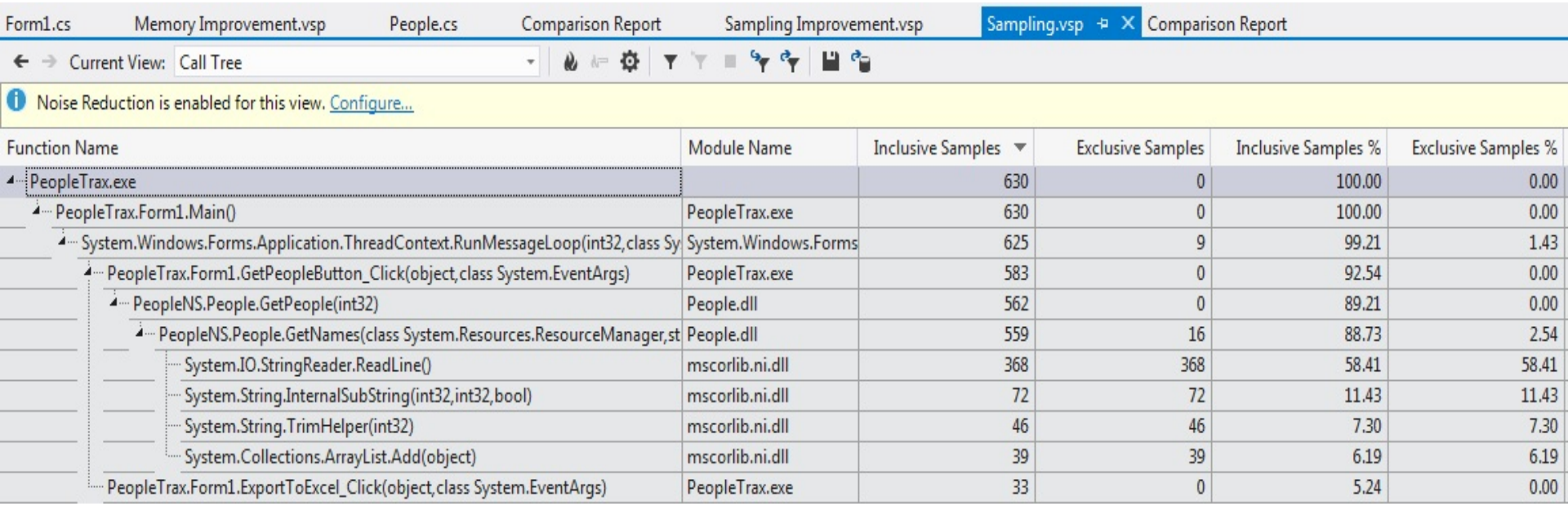

PeopleTrax130522.vsp

Current View: Functions

Function Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
System.IO.StringReader.ReadLine()	368	368	58.41	58.41
System.String.InternalSubString(int32,int32,bool)	72	72	11.43	11.43
System.String.TrimHelper(int32)	46	46	7.30	7.30
System.Collections.ArrayList.Add(object)	40	40	6.35	6.35
System.String.Concat(string,string)	28	28	4.44	4.44
System.Windows.Forms.ListView.ListViewItemCollectior	18	18	2.86	2.86
PeopleNS.People.GetNames(class System.Resources.Res	559	16	88.73	2.54
System.Resources.ResourceManager.GetString(string,cla	15	15	2.38	2.38
System.Windows.Forms.Application.ThreadContext.Run	625	9	99.21	1.43
[clr.dll]	4	4	0.63	0.63
System.String.CreateTrimmedString(int32,int32)	3	3	0.48	0.48
System.Windows.Forms.Control.SendMessage(int32,int3	3	3	0.48	0.48
System.Configuration.ConfigurationManager.get_AppS	2	2	0.32	0.32
System.Windows.Forms.ListView.ListViewNativeItemCol	2	2	0.32	0.32
System.Drawing.Color.FromName(string)	1	1	0.16	0.16
System.IO.TextWriter.WriteLine(string)	1	1	0.16	0.16

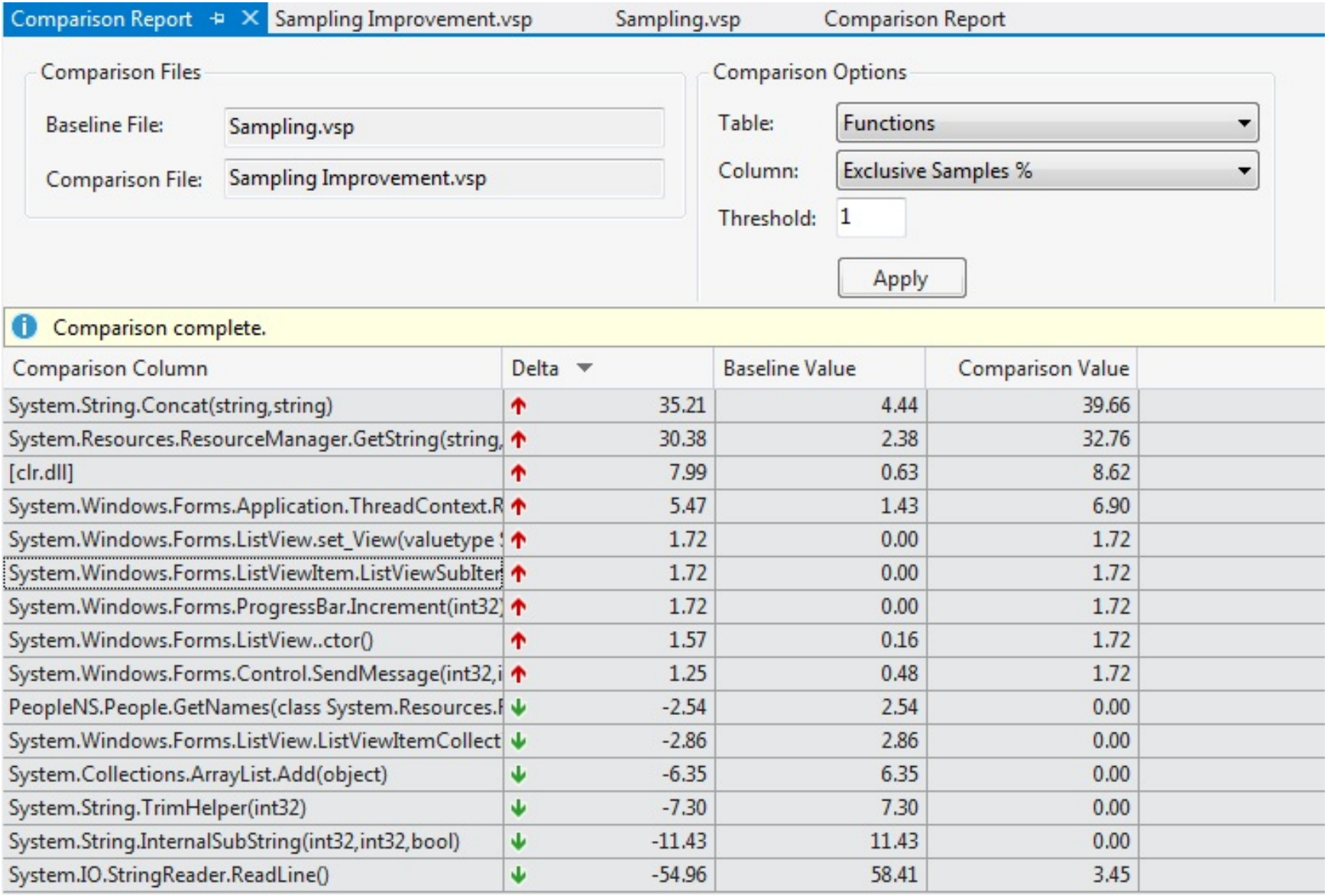
UI Representation

- Hierarchical Function Tree



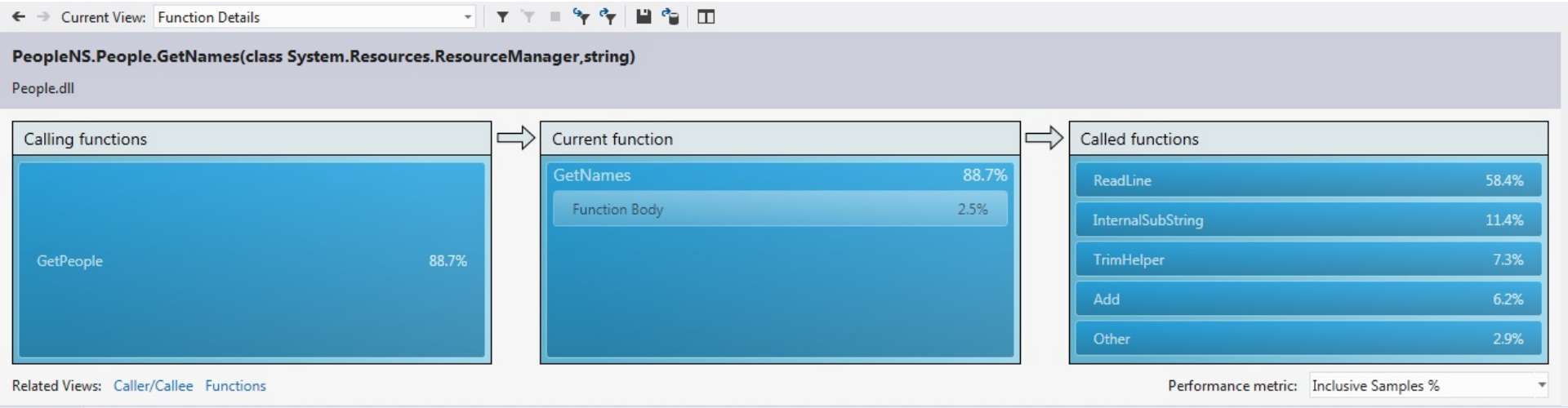
Function Name	Module Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
PeopleTrax.exe		630	0	100.00	0.00
PeopleTrax.Form1.Main()	PeopleTrax.exe	630	0	100.00	0.00
System.Windows.Forms.Application.ThreadContext.RunMessageLoop(int32,class Sy	System.Windows.Forms	625	9	99.21	1.43
PeopleTrax.Form1.GetPeopleButton_Click(object,class System.EventArgs)	PeopleTrax.exe	583	0	92.54	0.00
PeopleNS.People.GetPeople(int32)	People.dll	562	0	89.21	0.00
PeopleNS.People.GetNames(class System.Resources.ResourceManager,st	People.dll	559	16	88.73	2.54
System.IO.StringReader.ReadLine()	mscorlib.ni.dll	368	368	58.41	58.41
System.String.InternalSubString(int32,int32,bool)	mscorlib.ni.dll	72	72	11.43	11.43
System.String.TrimHelper(int32)	mscorlib.ni.dll	46	46	7.30	7.30
System.Collections.ArrayList.Add(object)	mscorlib.ni.dll	39	39	6.19	6.19
PeopleTrax.Form1.ExportToExcel_Click(object,class System.EventArgs)	PeopleTrax.exe	33	0	5.24	0.00

Comparison



Comparison Column	Delta	Baseline Value	Comparison Value
System.String.Concat(string,string)	35.21	4.44	39.66
System.Resources.ResourceManager.GetString(string,	30.38	2.38	32.76
[clr.dll]	7.99	0.63	8.62
System.Windows.Forms.Application.ThreadContext.R	5.47	1.43	6.90
System.Windows.Forms.ListView.set_View(valuetype	1.72	0.00	1.72
System.Windows.Forms.ListViewItem.ListViewSubIter	1.72	0.00	1.72
System.Windows.Forms.ProgressBar.Increment(int32)	1.72	0.00	1.72
System.Windows.Forms.ListView..ctor()	1.57	0.16	1.72
System.Windows.Forms.Control.SendMessage(int32,i	1.25	0.48	1.72
PeopleNS.People.GetNames(class System.Resources.F	-2.54	2.54	0.00
System.Windows.Forms.ListView.ListViewItemCollect	-2.86	2.86	0.00
System.Collections.ArrayList.Add(object)	-6.35	6.35	0.00
System.String.TrimHelper(int32)	-7.30	7.30	0.00
System.String.InternalSubString(int32,int32,bool)	-11.43	11.43	0.00
System.IO.StringReader.ReadLine()	-54.96	58.41	3.45

Direct Code Check



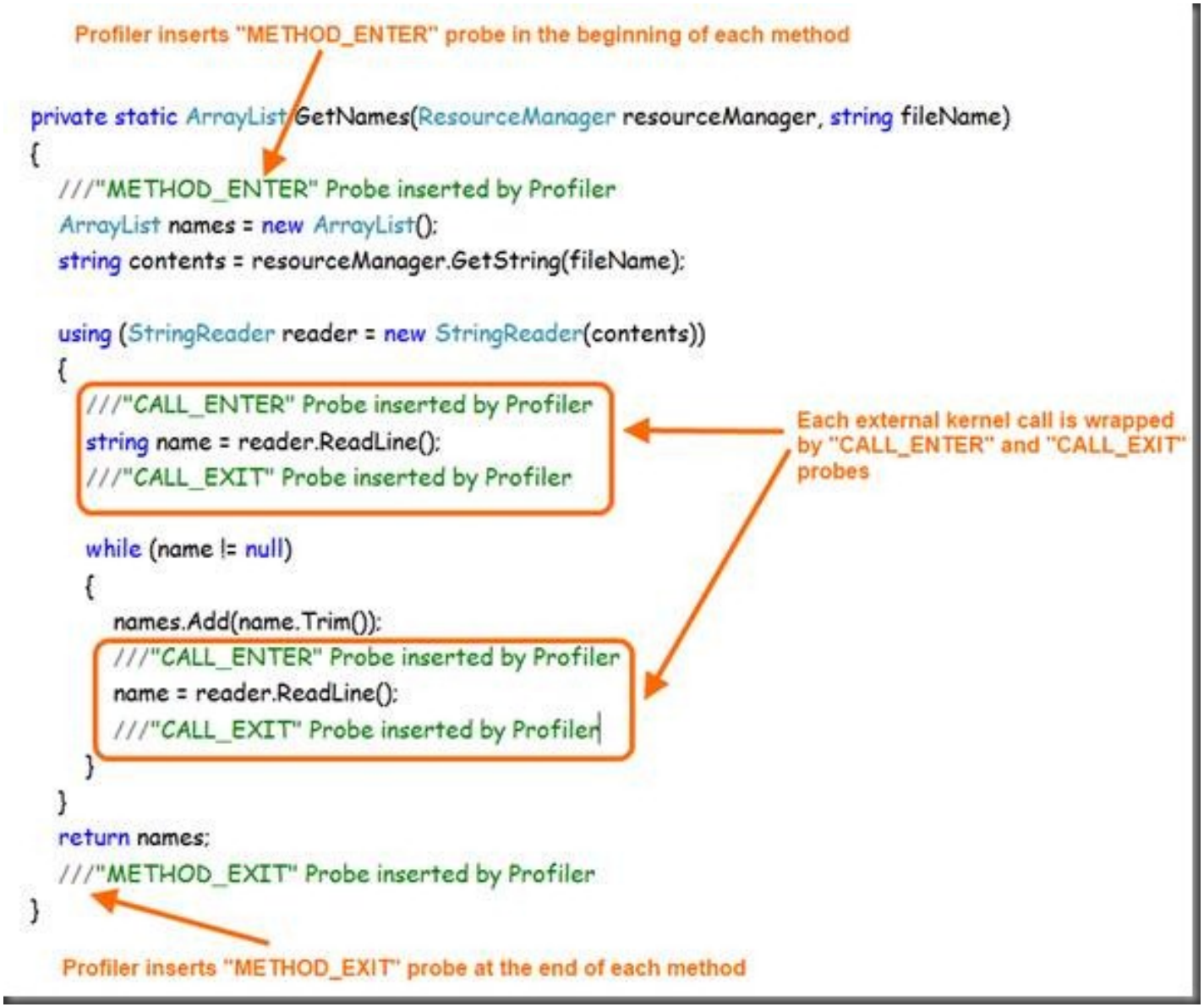

```
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Maintainability", "CA1500:VariableNamesShouldNotMatchFieldNames",
        private static ArrayList GetNames(ResourceManager resourceManager, string fileName)
        {
            ArrayList names = new ArrayList();
2.4 %       string contents = resourceManager.GetString(fileName);

            using (StringReader reader = new StringReader(contents))
            {
                string name = reader.ReadLine();

58.4 %          while (name != null)
                {
                    names.Add(name.Trim());
27.8 %              name = reader.ReadLine();
                }
            }
            return names;
        }
```

Instrumentation

- Injecting Probe to the Module (Intrusive)
- Capture timing information of each function call

Instrumentation

- Investigating I/O bottleneck such as Disk I/O
- Close Examination of Module
- Problem: Code bloat, slower performance


Profiler inserts "METHOD_ENTER" probe in the beginning of each method
private static ArrayList GetNames(ResourceManager resourceManager, string fileName)
{
///"METHOD_ENTER" Probe inserted by Profiler
ArrayList names = new ArrayList();
string contents = resourceManager.GetString(fileName);
using (StringReader reader = new StringReader(contents))
{
///"CALL_ENTER" Probe inserted by Profiler
string name = reader.ReadLine();
///"CALL_EXIT" Probe inserted by Profiler
while (name != null)
{
names.Add(name.Trim());
///"CALL_ENTER" Probe inserted by Profiler
name = reader.ReadLine();
///"CALL_EXIT" Probe inserted by Profiler
}
}
return names;
///"METHOD_EXIT" Probe inserted by Profiler
}
Each external kernel call is wrapped by "CALL_ENTER" and "CALL_EXIT" probes
Profiler inserts "METHOD_EXIT" probe at the end of each method

Instrumentation Representation

Instrumentation Function View Column Name	Definition
Elapsed Exclusive Values	The time that was spent executing a function, excluding time that was spent in child functions.
Application Exclusive Values	The time that was spent executing a function and its child functions, excluding time that was spent in operating system events.
Elapsed Inclusive Values	The total time that was spent executing a function and its child functions.
Application Inclusive Values	The time that was spent executing a function and its child functions, excluding time that was spent in operating system events.
Elapsed Inclusive/Exclusive Percent	100 * Function Elapsed (Exclusive/Inclusive) / Session Elapsed Inclusive
Application Inclusive/Exclusive Percent	100 * Function Application (Inclusive/Exclusive) / Session Application Inclusive

Function Name	Number of Calls	Elapsed Inclusive Time %	Elapsed Exclusive Time...	Avg Elapsed Inclusive Time	Avg Elapsed Exclusive Time	Module Name
System.Windows.Forms.Application.Run(class	1	99.91	44.26	94,341.49	41,786.95	System.Windows.Forms.dll
PeopleNS.People.GetNames(class System.Resc	3,000	55.03	25.51	17.32	8.03	People.dll
System.IO.TextReader.ReadLine()	15,003,000	12.21	12.21	0.00	0.00	mscorlib.dll
System.String.Trim()	15,000,000	9.19	9.19	0.00	0.00	mscorlib.dll
System.Collections.ArrayList.Add(object)	15,001,500	8.03	8.03	0.00	0.00	mscorlib.dll
ListViewItemCollection.Add(class System.Winc	1,500	0.35	0.35	0.22	0.22	System.Windows.Forms.dll
System.String.Concat(string,string)	6,000	0.12	0.12	0.02	0.02	mscorlib.dll
System.Resources.ResourceManager.GetString	3,000	0.07	0.07	0.02	0.02	mscorlib.dll
OperationNS.OperationControl.Increment(int3	1,500	0.04	0.04	0.02	0.02	Operation.dll
PeopleTrax.Form1..ctor()	1	0.07	0.03	68.10	30.23	PeopleTrax.exe
System.Diagnostics.Process.Start()	1	0.03	0.03	25.44	25.44	System.dll
System.Windows.Forms.Form.Dispose(bool)	1	0.02	0.02	23.42	23.36	System.Windows.Forms.dll
System.Configuration.ConfigurationManager.	2	0.01	0.01	4.81	4.81	System.Configuration.dll
OperationNS.OperationControl.Increment(int3	3,000	0.01	0.01	0.00	0.00	Operation.dll
System.Windows.Forms.ListView..ctor()	1	0.01	0.01	8.94	8.94	System.Windows.Forms.dll
OperationNS.OperationControl..ctor()	1	0.01	0.01	8.74	8.02	Operation.dll
OperationNS.OperationControl..ctor()	1	0.01	0.01	7.85	7.49	Operation.dll
PeopleNS.People.GetPeople(int32)	1	55.09	0.01	52,012.51	7.48	People.dll
OperationNS.OperationControl.Start(string,int	2	0.01	0.01	5.39	3.67	Operation.dll
PeopleTrax.Form1.ExportToExcel_Click(object,	1	0.17	0.01	156.65	6.54	PeopleTrax.exe
PeopleNS.People.GetPeople(int32)	1	55.08	0.01	52,005.02	5.27	People.dll

Sampling vs Instrumentation

Sampling	*Instrumentation*
Nonintrusive (Collect Function CallStack)	**Intrusive (Probe Function)**
Low Overhead	**Higher Runtime Overhead**
Track Application actively using CPU blind to I/O, sleep, or other blocked time.	**Disk I/O, network, Database I/O, other external Resources**

Memory Profiling

- Collect Information about size and number of objects that were created or destroyed in garbage collection (.NET CLR).

Memory Event

- Interrupt Processor each Allocation .NET Object allocation or lifetime creation
- Data is generated using Sampling and Instrumentation

Functions Allocating Most Memory

Name	Bytes %
PeopleNS.People.GetPeople(int32)	84.32
System.String.Concat(string,string)	15.56
ListViewItemCollection.Add(class System.Windows.Forms.ListViewItem)	0.03
System.Windows.Forms.ListViewItem..ctor(string[])	0.02
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	0.02

Types With Most Memory Allocated

Name	Bytes %
System.String	80.30
System.Object[]	19.56
System.Char[]	0.03
System.Collections.Hashtable.bucket[]	0.01
System.Windows.Forms.ListViewItem	0.01

Types With Most Instances

Name	Instances %
System.String	99.63
System.Object[]	0.17
System.Int32	0.03
System.Collections.ArrayList	0.01
System.IO.StringReader	0.01

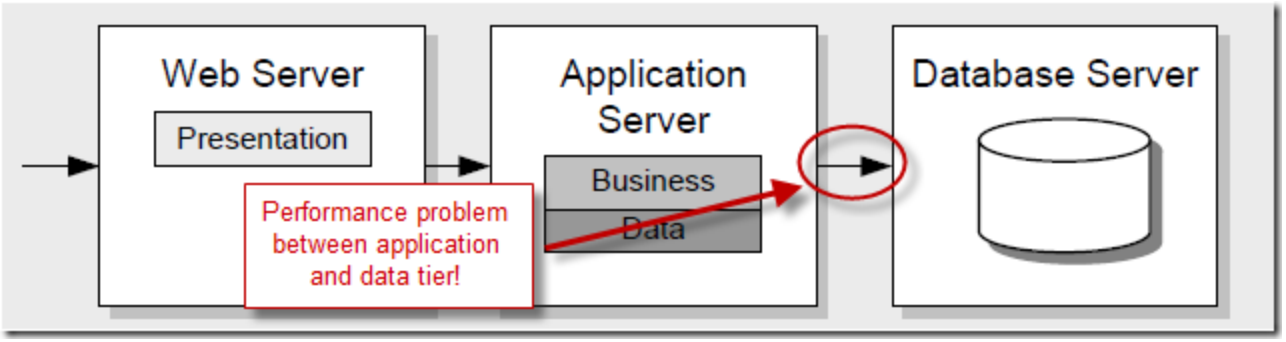
Tier Interaction

- Monitor Database Layer on the Application
- Monitor Encapsulated ADO.NET query (EntityFramework, NHibernate)

EntityFramework

- Object Relational Mapping in .Net
- Domain Specific Object and Properties
- Linq or Entity Query

Example:
employees.**Where**(e => e.DepartmentId == 5)



EF Representation

Current View: Tier Interactions

Name	Database	Count	Total Elapsed T...	Min Elapsed Time	Max Elapsed Time	Avg Elapsed Time
WebDev.WebServer40.EXE						
Database Connections						
EASTEN-JIJIKZ	prodinner	73 Queries	154.78			

Database connection details:

Command Text	Query Calls	Total Elapsed T...	Min Elapsed Time	Max Elapsed Time	Avg Elapsed Time
SELECT TOP (10) [Project1].[Id] AS [Id], [Project1].[Name] AS [Name], [Project1].[CountryId] AS [C	1	45.36	45.36	45.36	45.36
SELECT [Project1].[Id] AS [Id], [Project1].[Name] AS [Name], [Project1].[CountryId] AS [CountryId]	2	34.86	0.91	33.95	17.43
SELECT [Extent1].[Id] AS [Id], [Extent1].[FirstName] AS [FirstName], [Extent1].[LastName] AS [LastN	2	34.07	0.39	33.68	17.03
update [dbo].[Dinners]set [IsDeleted] = @0where ([Id] = @1)	2	17.07	0.22	16.85	8.53
SELECT [Extent2].[Id] AS [Id], [Extent2].[Name] AS [Name], [Extent2].[Comments] AS [Comments],	24	6.75	0.17	0.66	0.28
SELECT [Extent1].[Id] AS [Id], [Extent1].[Name] AS [Name], [Extent1].[IsDeleted] AS [IsDeleted]FROI	28	6.18	0.13	0.40	0.22
SELECT [Extent1].[Id] AS [Id], [Extent1].[FirstName] AS [FirstName], [Extent1].[LastName] AS [LastN	9	5.03	0.15	2.79	0.56
SELECT TOP (2) [Extent1].[Id] AS [Id], [Extent1].[Name] AS [Name], [Extent1].[CountryId] AS [Coun	2	4.60	1.03	3.57	2.30
SELECT [GroupBy1].[A1] AS [C1]FROM (SELECT COUNT(1) AS [A1]FROM [dbo].[Dinners] AS [Exte	3	0.86	0.26	0.33	0.29

Contention Concurrency

- Profiling competing thread (accessed shared resource)
- Help Identifying Lock Contention during Multithreading

Resource

- http://msdn.microsoft.com/en-us/library/z9z62c29.aspx
- http://blogs.msdn.com/b/dalaqab/archive/2009/12/22/q-which-profiling-mode-should-i-use.aspx

DEMO

- http://nopcommerce.codeplex.com/SourceControl/changeset/38dec9e1c22e (Tier Interaction)
- http://nopcommerce.codeplex.com/SourceControl/changeset/c12d27cec7cc (Less IO)