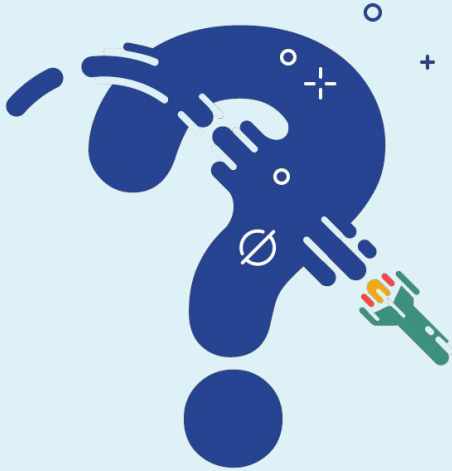
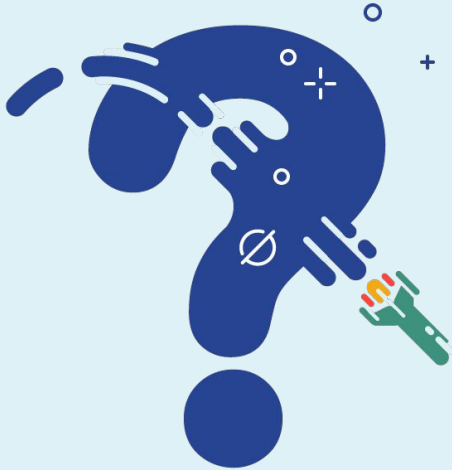


Ask Any Questions During the Talk



Are You Wondering
What Infinite
Improbability Is?



Infinite Improbability Testing- Execute All Tests in Parallel

Anton Angelov

Co-founder, Chief Development

AUTOMATE THE PLANET

Site: <http://bellatrix.solutions>

Blog: <http://automatetheplanet.com>

@angelovstanton

aangelov@bellatrix.solutions



Agenda

Problem

Parallel Test Execution?

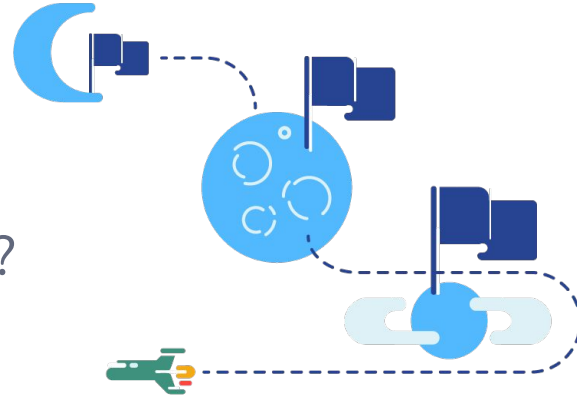
Advantages

How People Run 'Heavy' Tests?

Meissa

Challenges & Solutions

Go Even Further



The Problem

Delivery cycles will get even shorter. More than half of respondents want to have shorter delivery cycles in the next two years. Sixty-eight percent (68%) want to deploy weekly, daily, or multiple times a day, as compared to the 41% who do so already.

The State of Testing Report for 2017

BY SMARTBEAR

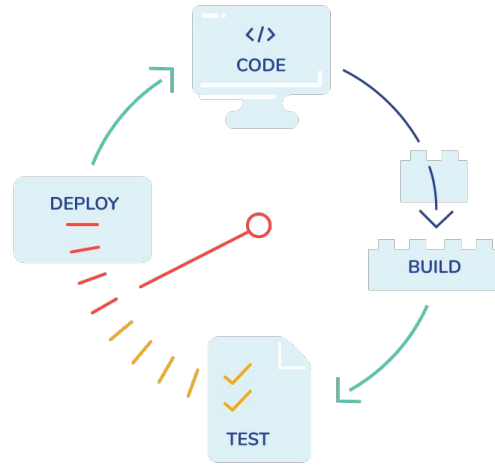
The Problem

Delivery cycles will get even shorter. More than half of respondents want to have shorter delivery cycles in the next two years. Sixty-eight percent (68%) want to deploy weekly, daily, or multiple times a day, as compared to the 41% who do so already.

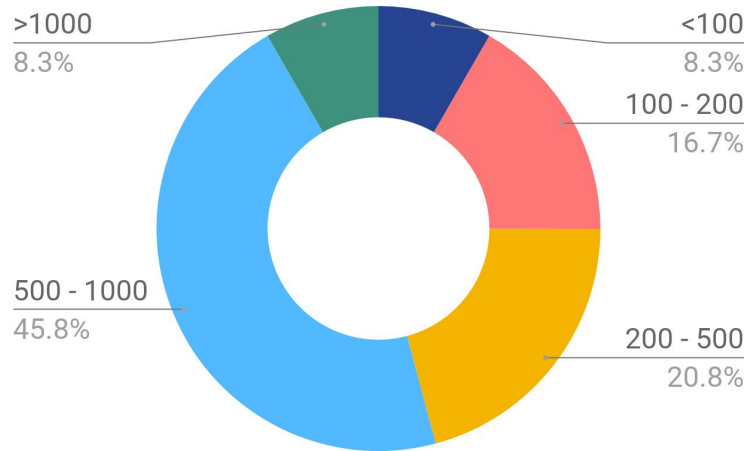
The State of Testing Report for 2017

BY SMARTBEAR

How Do People Run Their 'Heavy' Tests?



How Many Tests Do You Have?



750

AVERAGE TESTS COUNT

6h

AVERAGE TEST EXECUTION

3h

AVERAGE TEST EXECUTION

When Do You Execute Your Tests?

CI- All of Them	42.85%
CI- Some, Nightly- All	28.57%
CI- Some, Occasionally- Rest	10.71%
No CI, Nightly All	7.14%
Other	10.71%

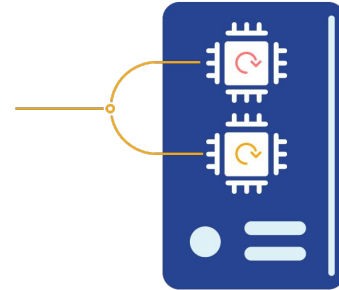
60%

STRUGGLE WITH CI

What Is a Parallel Test Execution?

Parallel Testing

Single Machine - Multiple Processes



What Is a Parallel Test Execution?

Distributed Testing

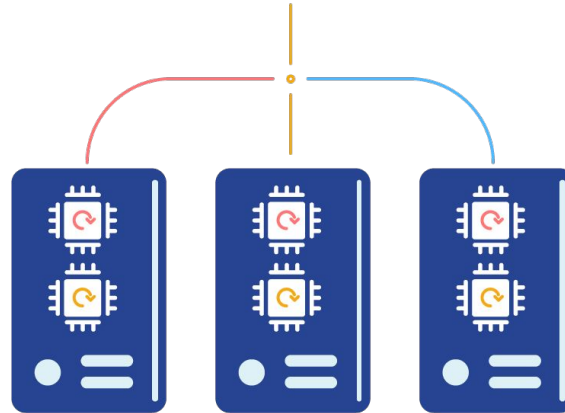
Multiple Machines - Single Process



What Is a Parallel Test Execution?

Parallel Distributed Testing

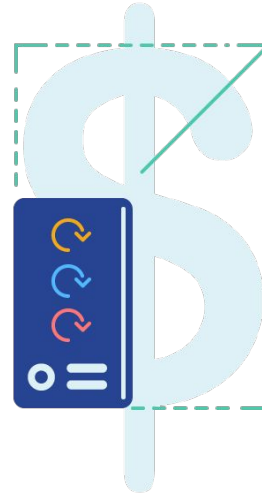
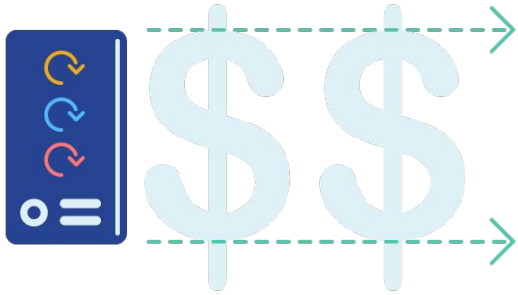
Multiple Machines -
Multiple Processes



Why Do You Need to Distribute Your Tests?

$$1.5-2.0 \times \text{CPU Cores} = \text{PT}$$

One Big Machine or Many Smaller?



Advantages



Speed



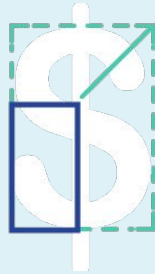
Continuous
Delivery



Continuous
Integration



Higher
Coverage



Increased
ROI



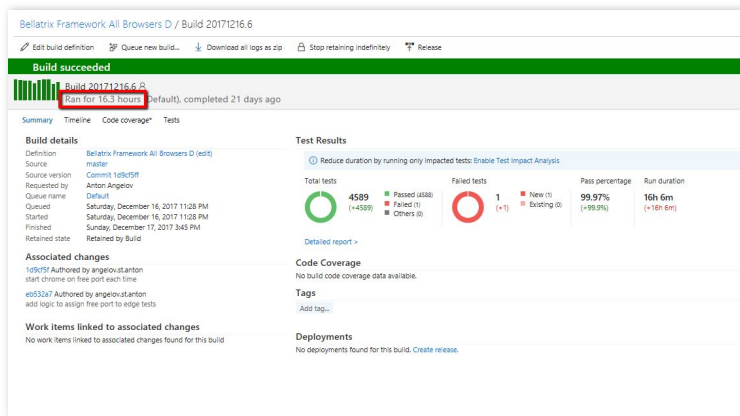
Locate Flaky
Tests

The Bellatrix Story

Bellatrix is a cross-platform, an easily customizable and extendable .NET test automation framework that increases tests' reliability.



The Bellatrix Story



4500

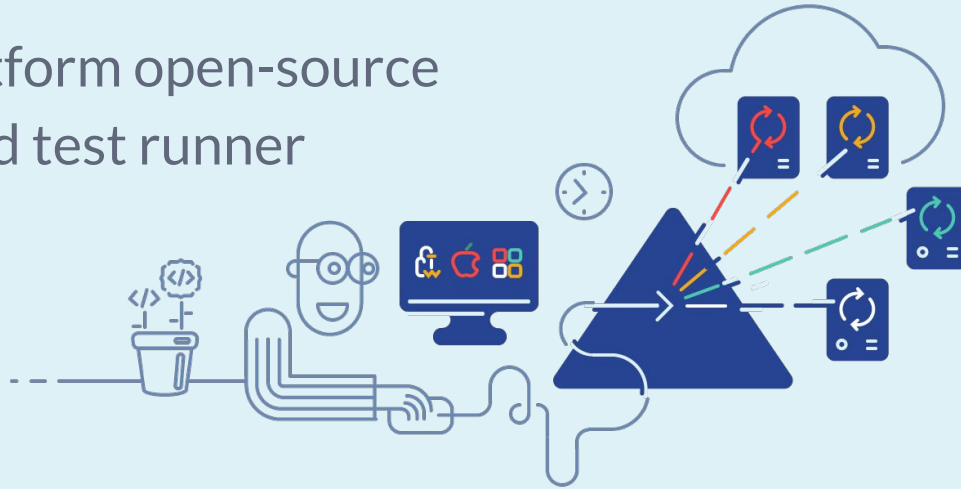
UI TESTS

16h

TEST EXECUTION

Meissa

Cross-platform open-source
distributed test runner



<https://meissarunner.com>

Features



Parallel on Agent



Smart Balancing



Failing Test Retry



Single CLI

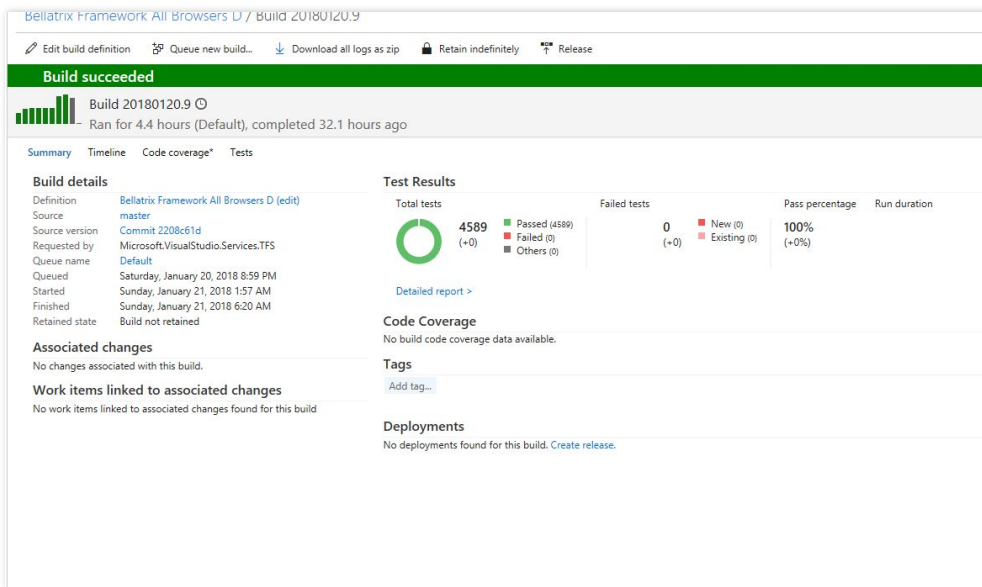


Abortion of Runs



Cross-platform

After Meissa



4500

UI TESTS

16h

NORMAL TEST EXECUTION

4h

MEISSA TEST EXECUTION

How Do We Use It?

1

Start Meissa Server

```
meissa.exe initServer
```

How Do We Use It?

2 Start Meissa Test Agent

```
meissa.exe testAgent --testAgentTag="APIAgent"  
--testServerUrl="http://IPServerMachine:5000"
```

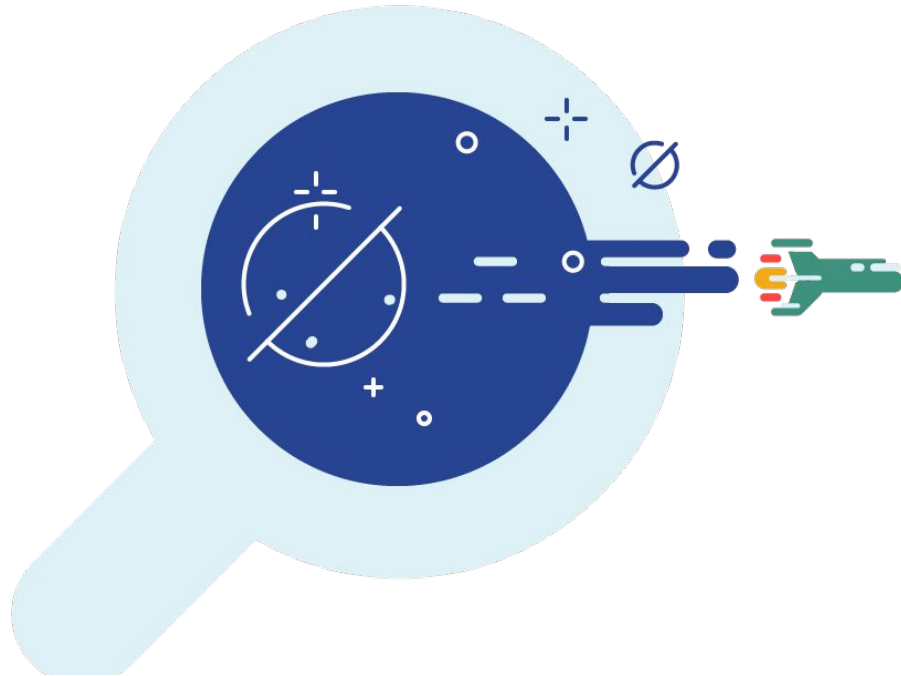
How Do We Use It?

3

Start Meissa Tests Runner

```
meissa.exe runner --resultsFilePath="pathToResults\result.trx"  
--outputFilesLocation="pathToBuildedFiles"  
--agentTag="APIAgent" --testTechnology="MSTestCore"  
--testLibraryPath="pathToBuildedFiles\SampleTestProj.dll"
```

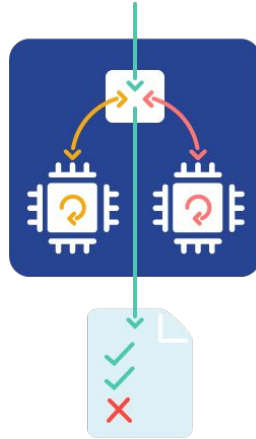
Space Exploration



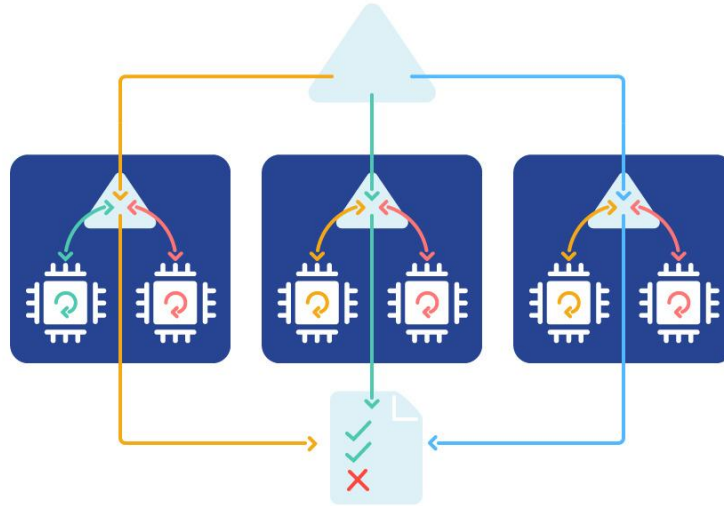
How Do You Run Your Tests in Parallel?

Unit Test Framework + Selenium Grid	56.25%
CI Tool as Jobs and Slaves	18.75%
Microsoft Test Agents	6.25%
Custom Solution	18.75%

Native Runners



Native Runners + Meissa



Comparison

	Selenium	Meissa
WEB	w/o Edge, IE, Safari	✓
MOBILE & DESKTOP		✓
PARALLEL	✓	✓
DISTRIBUTING		✓
RETRY		✓
BALANCING		✓
CODELESS		✓

Comparison

	Selenium Grid	Meissa
WEB	✓	✓
MOBILE & DESKTOP	✓	✓
PARALLEL	✓ *	✓
DISTRIBUTING		✓
RETRY		✓
BALANCING		✓
CODELESS		✓

Comparison

	CI Tool	Meissa
WEB	✓	✓
MOBILE & DESKTOP	✓	✓
PARALLEL	✓ *	✓
DISTRIBUTING	✓	✓
RETRY		✓
BALANCING	Manual	✓
CODELESS		✓

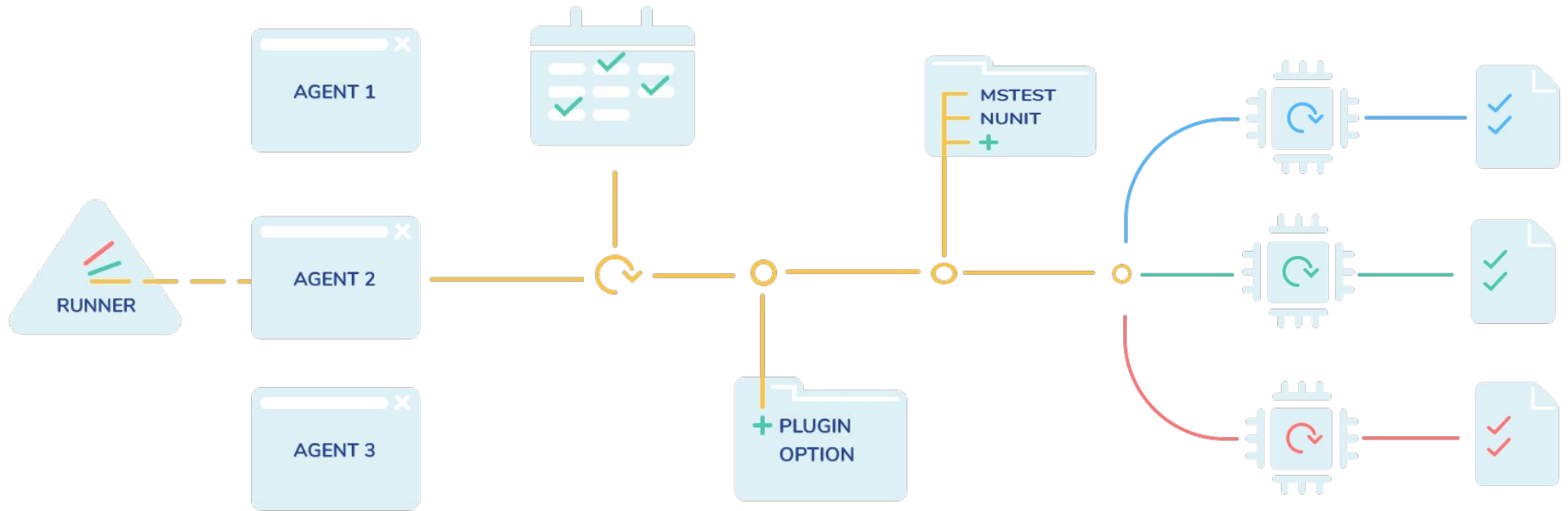
Comparison

	Microsoft MS Test Agents	Meissa
EASY SETUP & TROUBLESHOOT		✓
CANCEL		✓
PARALLEL	✓	✓
DISTRIBUTING	✓	✓
RETRY		✓
BALANCING	Hard to configure	✓
CODELESS	✓	✓

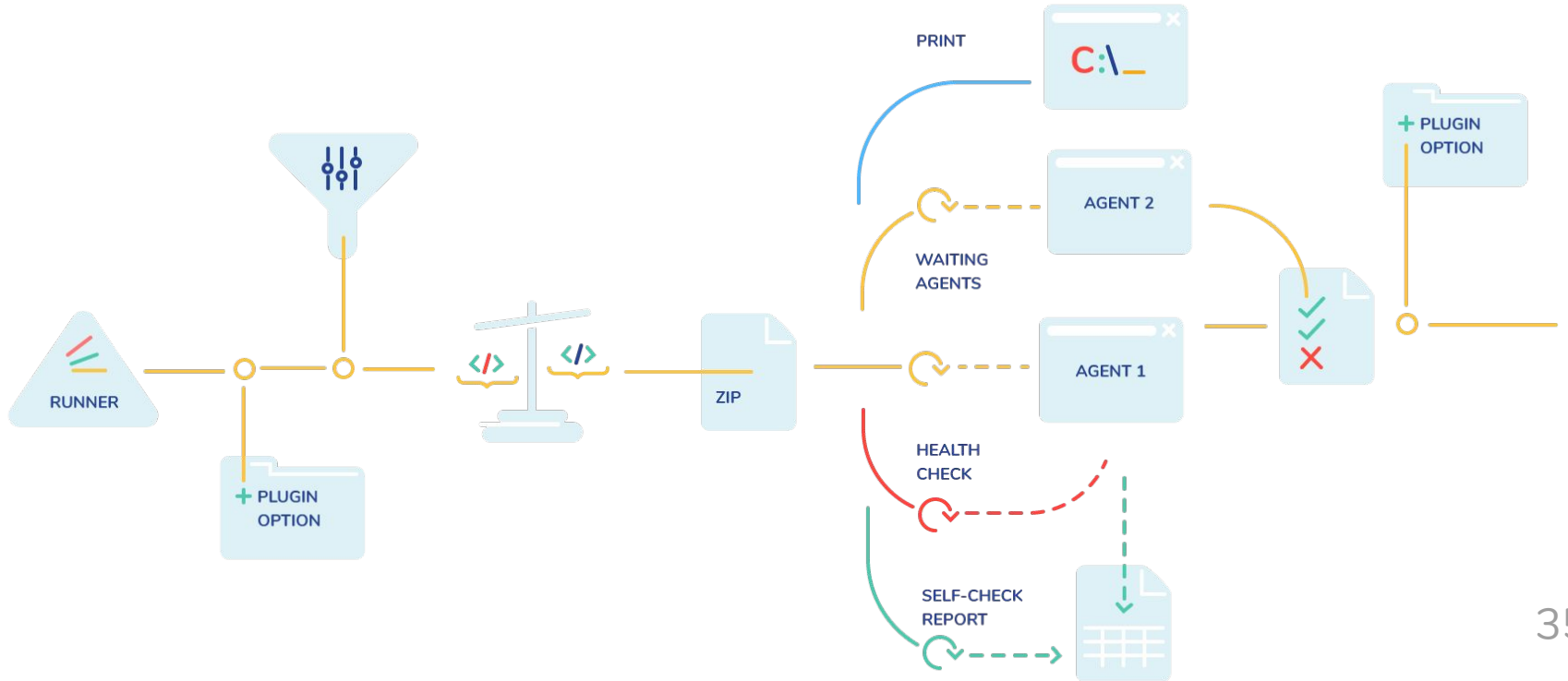
How It Works Internally?



Meissa Test Agent Mode

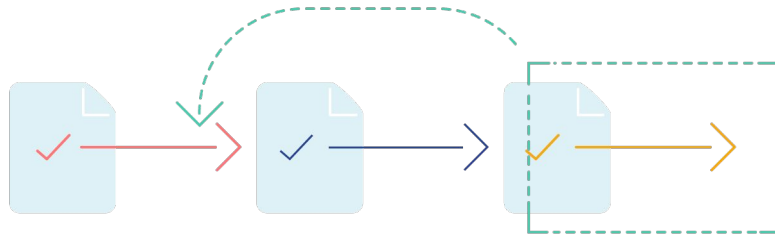


Meissa Test Runner Mode



Challenges Running Tests in Parallel

Preparing Environment and Managing Data



Execute the expensive setup, cleanup before/after the run.
Each test uses its own unique set of data.

Keep The Clean State of Agents



```
public static void Dispose()
{
    var processes = Process.GetProcesses();
    var processesToCheck = new List<string> { "edge", "chrome", "firefox" };
    foreach (var process in processes)
    {
        if (processesToCheck.Contains(process.ProcessName.ToLower()))
        {
            process.Kill();
            process.WaitForExit();
        }
    }
}
```

Video Recording



Use separate agents one browser at a time

Screenshots



Create full page screenshots with JavaScript

Port Already in Use



```
private static int GetFreeTcpPort()  
{  
    var tcpListener = new TcpListener(IPAddress.Loopback, 0);  
    tcpListener.Start();  
    int port = ((IPEndPoint)tcpListener.LocalEndPoint).Port;  
    tcpListener.Stop();  
    return port;  
}
```

Edge Driver Cannot Start Multiple Instances



Use separate agents one browser at a time

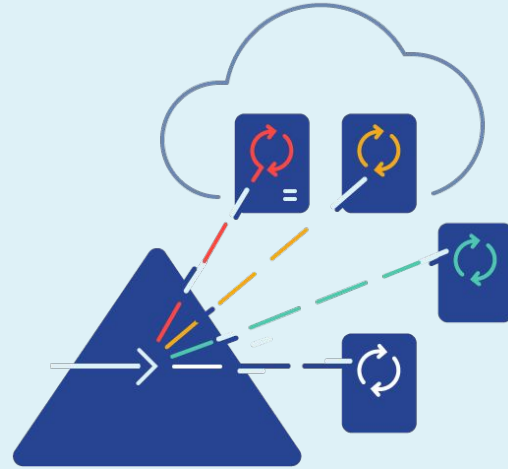
When It Is Appropriate to Use and When Not?

- Have Only Unit Tests
- Happy with Current Solution



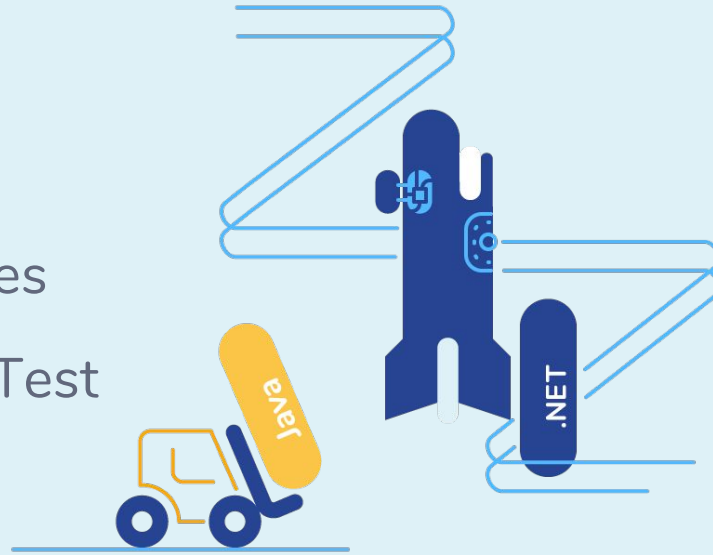
Getting Started

1. Go to meissarunner.com
2. Unzip It
3. Start Executing
4. Check the Documentation



Get Involved

- Report Bugs
- Suggest New Features
- Help Us Write More Test Runners
- Go to GitHub and Send a Message

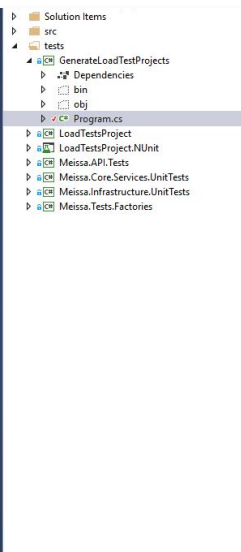


Summary

- Parallel Testing?
- Different Types
- Distributed Test Execution - Benefits
- Different Styles Running Tests in Parallel
- The Meissa Project
- How We Use It
- Why We Built Meissa

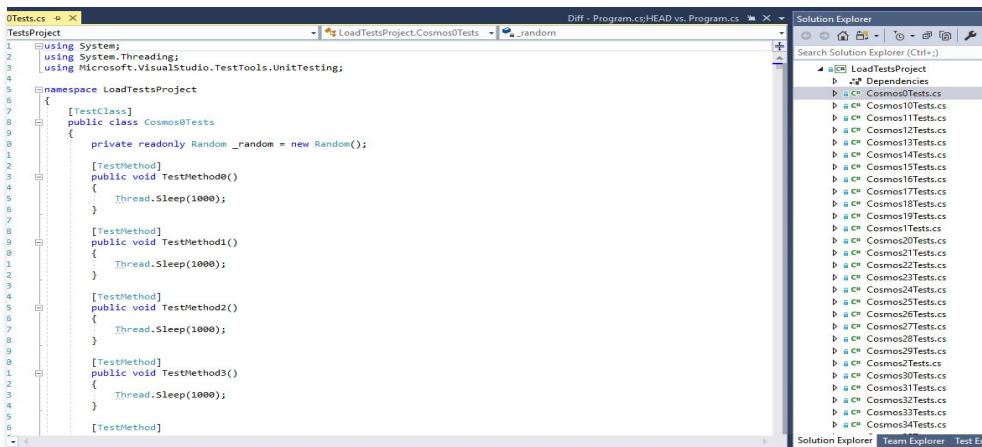
The Biggest Experiment

```
4 namespace GenerateLoadTestProjects
5 {
6     public class Program
7     {
8         public static void Main(string[] args)
9         {
10             GenerateLoadTests%stest();
11         }
12
13         private static void GenerateLoadTestsNUnit()
14         {
15             int numberOfTestFiles = 10;
16             int numberOfTests = 100;
17             string deployLocation = @"D:\OneDrive\SourceCode\AutomateThePlanet\Meissa\LoadTestsProject\NUnit";
18             for (int i = 0; i < numberOfTestFiles; i++)
19             {
20                 var sb = new StringBuilder();
21                 sb.AppendLine("using System;");
22                 sb.AppendLine("using System.Threading;");
23                 sb.AppendLine("using NUnit.Framework;");
24                 sb.AppendLine();
25                 sb.AppendLine("namespace LoadTestsProject");
26                 sb.AppendLine("{");
27                 sb.AppendLine("    [TestFixture]");
28                 sb.AppendLine($"    public class Cosmos{i}Tests");
29                 sb.AppendLine("    {");
30                 sb.AppendLine($"        private readonly Random _random = new Random();");
31                 for (int j = 0; j < numberOfTests; j++)
32                 {
33                     sb.AppendLine("        ");
34                     sb.AppendLine("        [Test]");
35                     sb.AppendLine($"        public void TestMethod{j}());");
36                     sb.AppendLine("        {");
37                     sb.AppendLine("            Thread.Sleep(1000);");
38                     sb.AppendLine("        }");
39                 }
40                 sb.AppendLine("    }");
41                 sb.AppendLine("}");
42                 string fileLocation = Path.Combine(deployLocation, $"Cosmos{i}Tests.cs");
43                 File.WriteAllText(fileLocation, sb.ToString());
44             }
45         }
46     }
47 }
```



The image shows a Visual Studio code editor with a C# file named Program.cs. The code defines a Program class with a Main method that calls GenerateLoadTests%stest(). It also has a private static method GenerateLoadTestsNUnit() that generates test files. The code uses a StringBuilder to build the content of each test file, including using statements, namespace declarations, class declarations, and test methods. The test methods use Thread.Sleep(1000) to simulate work. The files are saved to a specific directory. On the right side, the Solution Explorer shows a project structure with folders for src, tests, and sub-projects like GenerateLoadTestProjects, Dependencies, bin, obj, and Program.cs. Under Program.cs, there are several test projects listed, including LoadTestsProject, LoadTestsProject.NUnit, Meissa.API.Tests, Meissa.Core.Services.UnitTests, Meissa.Infrastructure.UnitTests, and Meissa.Tests.Factories.

The Biggest Experiment



```
1 using System;
2 using System.Threading;
3 using Microsoft.VisualStudio.TestTools.UnitTesting;
4
5 namespace LoadTestsProject
6 {
7     [TestClass]
8     public class Cosmos0Tests
9     {
10         private readonly Random _random = new Random();
11
12         [TestMethod]
13         public void TestMethod0()
14         {
15             Thread.Sleep(1000);
16         }
17
18         [TestMethod]
19         public void TestMethod1()
20         {
21             Thread.Sleep(1000);
22         }
23
24         [TestMethod]
25         public void TestMethod2()
26         {
27             Thread.Sleep(1000);
28         }
29
30         [TestMethod]
31         public void TestMethod3()
32         {
33             Thread.Sleep(1000);
34         }
35
36         [TestMethod]
37         // ...
38     }
39 }
```

Solution Explorer

- LoadTestsProject
- Dependencies
- Cosmos0Tests.cs
- Cosmos10Tests.cs
- Cosmos11Tests.cs
- Cosmos12Tests.cs
- Cosmos13Tests.cs
- Cosmos14Tests.cs
- Cosmos15Tests.cs
- Cosmos16Tests.cs
- Cosmos17Tests.cs
- Cosmos18Tests.cs
- Cosmos19Tests.cs
- Cosmos1Tests.cs
- Cosmos20Tests.cs
- Cosmos21Tests.cs
- Cosmos22Tests.cs
- Cosmos23Tests.cs
- Cosmos24Tests.cs
- Cosmos25Tests.cs
- Cosmos26Tests.cs
- Cosmos27Tests.cs
- Cosmos28Tests.cs
- Cosmos29Tests.cs
- Cosmos2Tests.cs
- Cosmos30Tests.cs
- Cosmos31Tests.cs
- Cosmos32Tests.cs
- Cosmos33Tests.cs
- Cosmos34Tests.cs

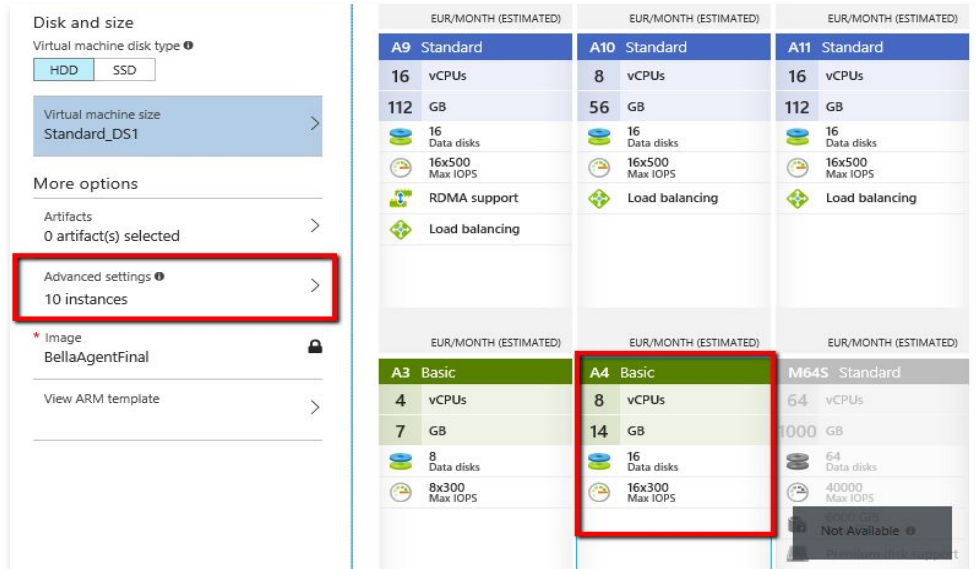
100000


HEAVY TESTS

28h

SEQUENTIAL ON SINGLE
MACHINE

The Biggest Experiment





Disk and size
Virtual machine disk type 
 HDD SSD

Virtual machine size
Standard_DS1 >


More options

Artifacts
0 artifact(s) selected >

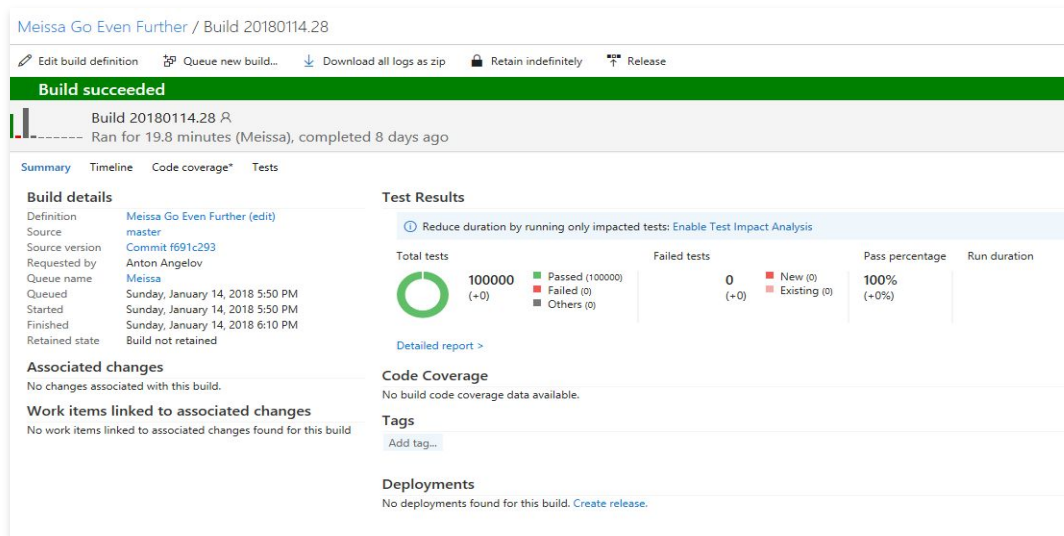
Advanced settings 
10 instances >

* Image
BellaAgentFinal 

View ARM template >

EUR/MONTH (ESTIMATED)		EUR/MONTH (ESTIMATED)		EUR/MONTH (ESTIMATED)	
A9 Standard	A10 Standard	A11 Standard			
16 vCPUs	8 vCPUs	16 vCPUs			
112 GB	56 GB	112 GB			
16 Data disks	16 Data disks	16 Data disks			
16x500 Max IOPS	16x500 Max IOPS	16x500 Max IOPS			
RDMA support	Load balancing	Load balancing			
Load balancing					
EUR/MONTH (ESTIMATED)		EUR/MONTH (ESTIMATED)		EUR/MONTH (ESTIMATED)	
A3 Basic	A4 Basic	M64S Standard			
4 vCPUs	8 vCPUs	64 vCPUs			
7 GB	14 GB	1000 GB			
8 Data disks	16 Data disks	64 Data disks			
8x300 Max IOPS	16x300 Max IOPS	40000 Max IOPS			
		Not Available 			

The Biggest Experiment



100000

HEAVY TESTS

1666min

SEQUENTIAL ON SINGLE
MACHINE

20min

PARALLEL ON 10
VIRTUAL MACHINES

I will end with a quote.

A great tester gives programmers immediate feedback on what they did right and what they did wrong. Believe it or not, one of the most valuable features of a tester is providing positive reinforcement. There is no better way to improve a programmer's morale, happiness, and subjective sense of well-being than a La Marzocco Linea espresso machine to have dedicated testers who get frequent releases from the developers, try them out, and give negative and positive feedback.

Joel Spolsky

Stack Overflow's CEO

I will end with a quote.

A great tester gives programmers immediate feedback on what they did right and what they did wrong. Believe it or not, one of the **most valuable features of a tester** is providing positive reinforcement. There is no better way to improve a programmer's morale, happiness, and subjective sense of well-being than a La Marzocco Linea espresso machine to have dedicated testers who get frequent releases from the developers, try them out, and give negative and positive feedback.

Joel Spolsky

Stack Overflow's CEO

I will end with a quote.

A great tester gives programmers immediate feedback on what they did right and what they did wrong. Believe it or not, one of the most valuable features of a tester is providing positive reinforcement. There is no better way to improve a programmer's morale, happiness, and subjective sense of well-being than a La Marzocco Linea espresso machine to have dedicated testers who get frequent releases from the developers, try them out, and give negative and positive feedback.

Joel Spolsky

Stack Overflow's CEO

I will end with a quote.

A great tester gives programmers immediate feedback on what they did right and what they did wrong. Believe it or not, one of the most valuable features of a tester is providing positive reinforcement. There is no better way to improve a programmer's morale, happiness, and subjective sense of well-being than a La Marzocco Linea espresso machine to have dedicated testers who get frequent releases from the developers, try them out, and give negative and positive feedback.

Joel Spolsky

Stack Overflow's CEO

I will end with a quote.

A great tester gives programmers immediate feedback on what they did right and what they did wrong. Believe it or not, one of the most valuable features of a tester is providing positive reinforcement. There is no better way to improve a programmer's morale, happiness, and subjective sense of well-being than a La Marzocco Linea espresso machine to have dedicated testers who get frequent releases from the developers, try them out, and give negative and positive feedback.

Joel Spolsky

Stack Overflow's CEO

Questions

