Measuring Performance Improvements

Elton Stoneman

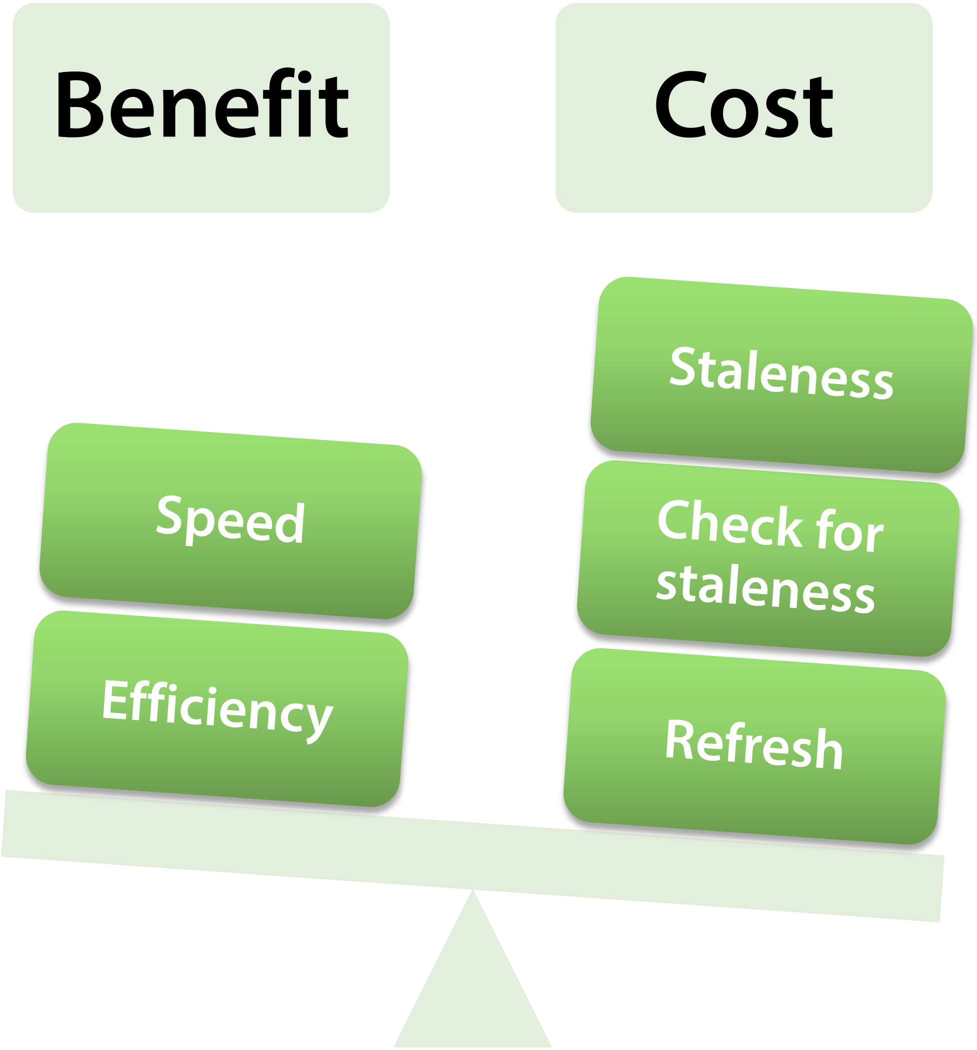
elton@sixeyed.com

pluralsight
hardcore developer training

Outline

- **Inspect & adapt**
- **At build time**
    - Automated tests
    - Unit tests
    - Integration tests
    - Performance tests
- **At run time**
    - Instrumentation
    - Performance counters

Inspect & Adapt



Benefit

Cost

Speed

Efficiency

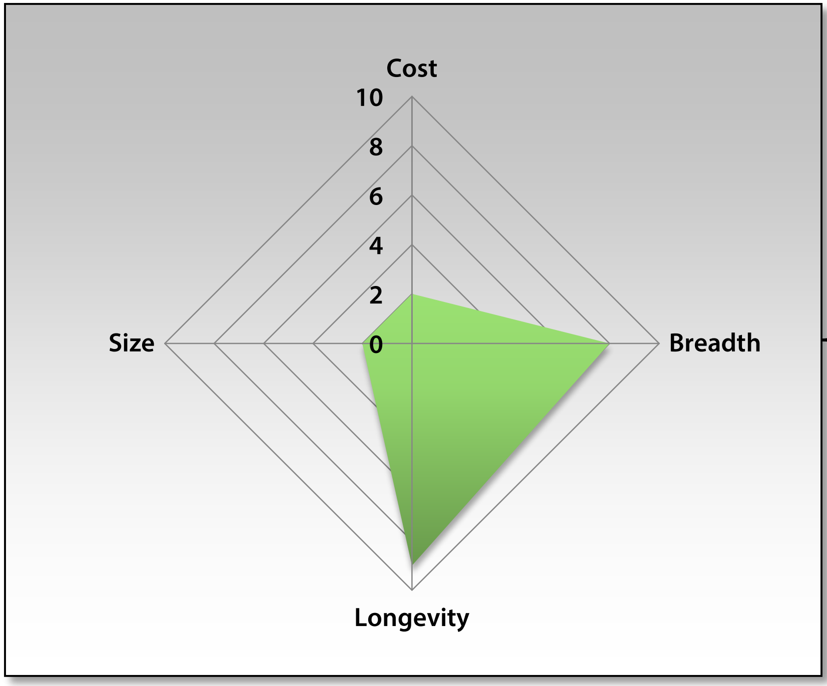
Staleness

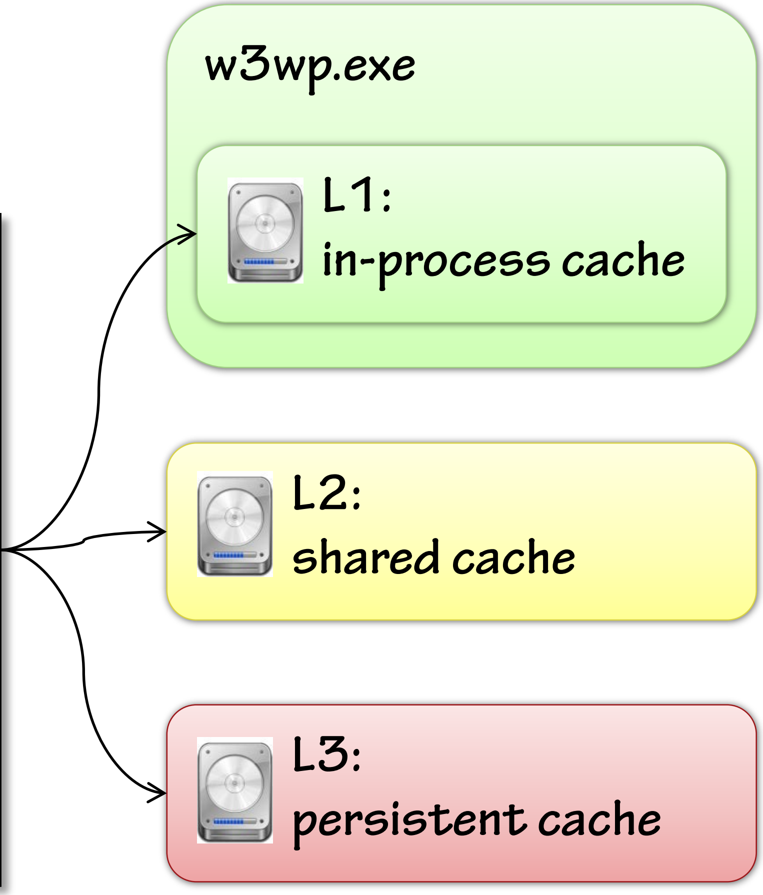
Check for staleness

Refresh

Inspect & Adapt

- **Solution caches**
  - □ Most effective store
  - □ Lifespan





w3wp.exe

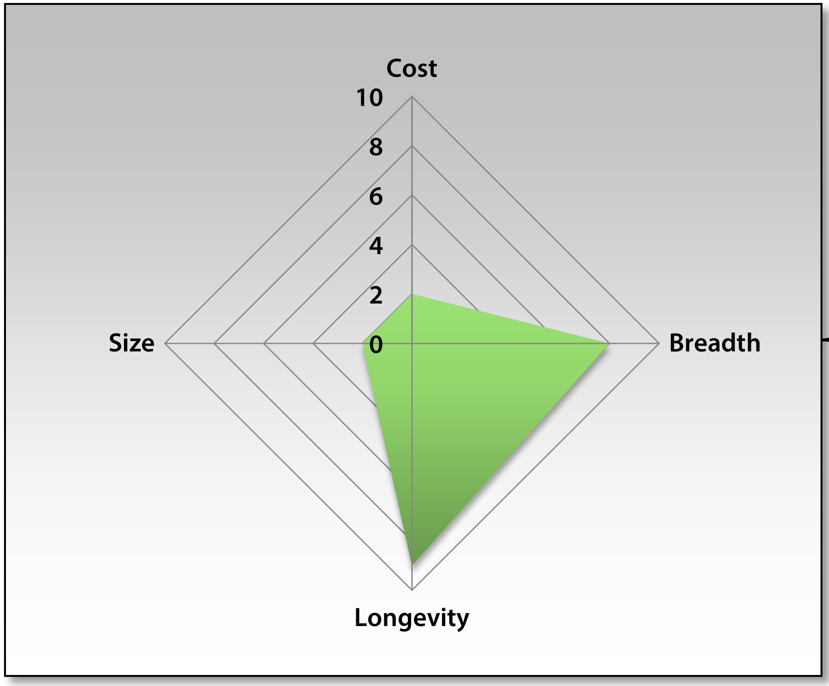
L1:
in-process cache

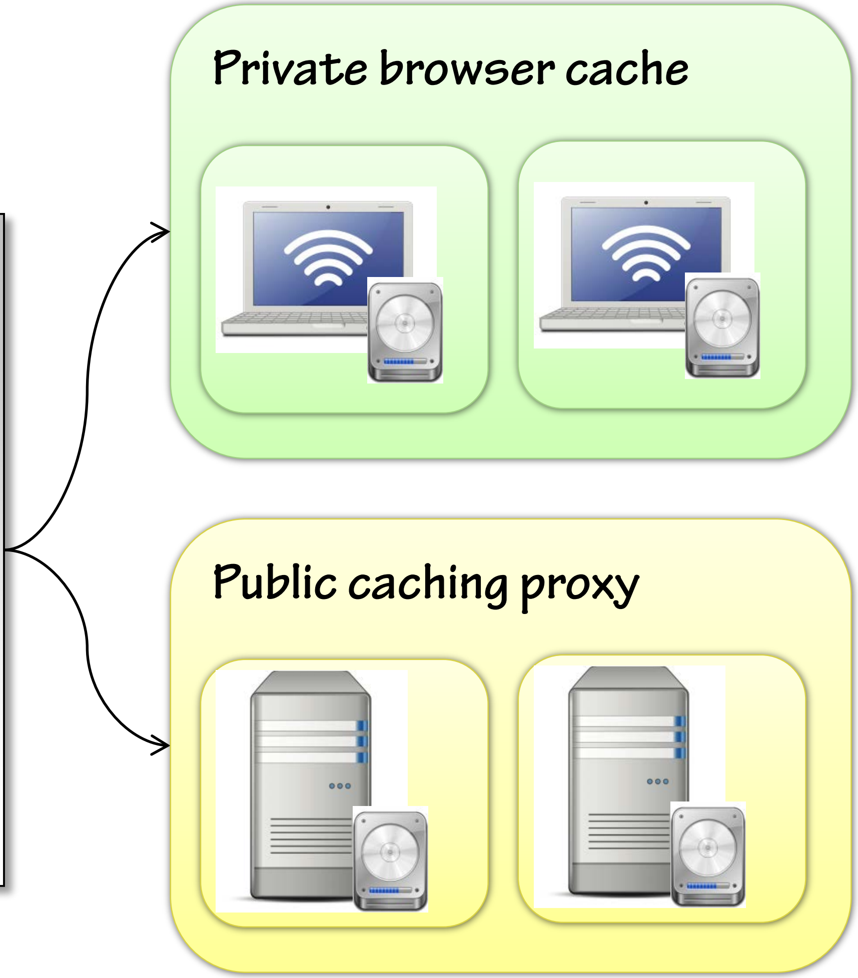
L2:
shared cache

L3:
persistent cache

Inspect & Adapt

- **External caches**
  - Invalidation and/or expiration
  - Lifespan





Private browser cache

Public caching proxy

Inspect & Adapt

- **Cache choices**
  - Informed guess
  - Business needs change
  - Data profiles change
- **Evaluate cache performance**
  - Ensure caches are working
  - Monitor effectiveness
  - React to changes

At Build Time

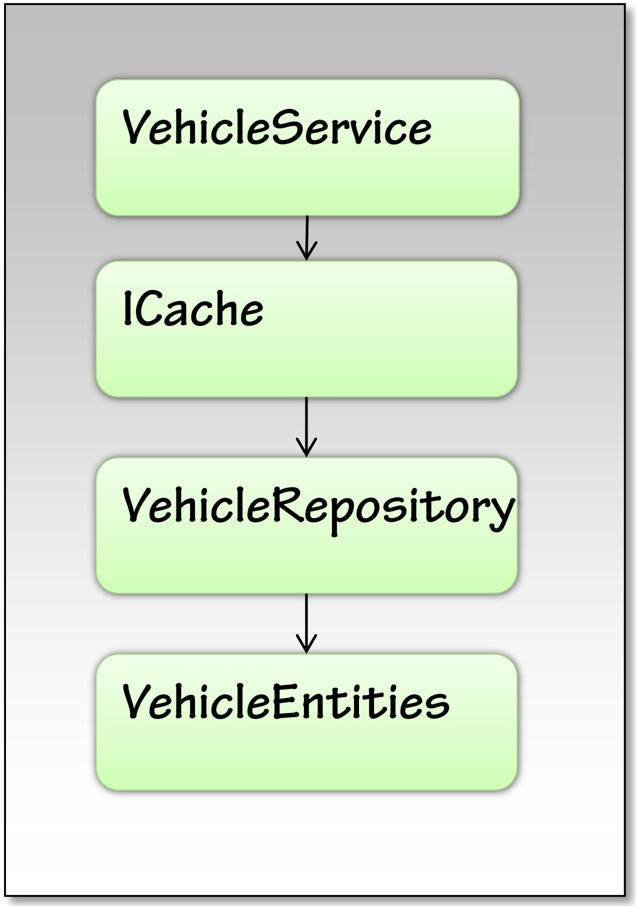
- **Solution caches**
  - Automated testing
- **Unit tests**
  - Mocking
  - Verify logic
- **Integration tests**
  - End-to-end web tests
  - Verify response & header correctness
- **Performance tests**
  - Speed
  - Capacity

At Build Time

- Demo: Unit Tests

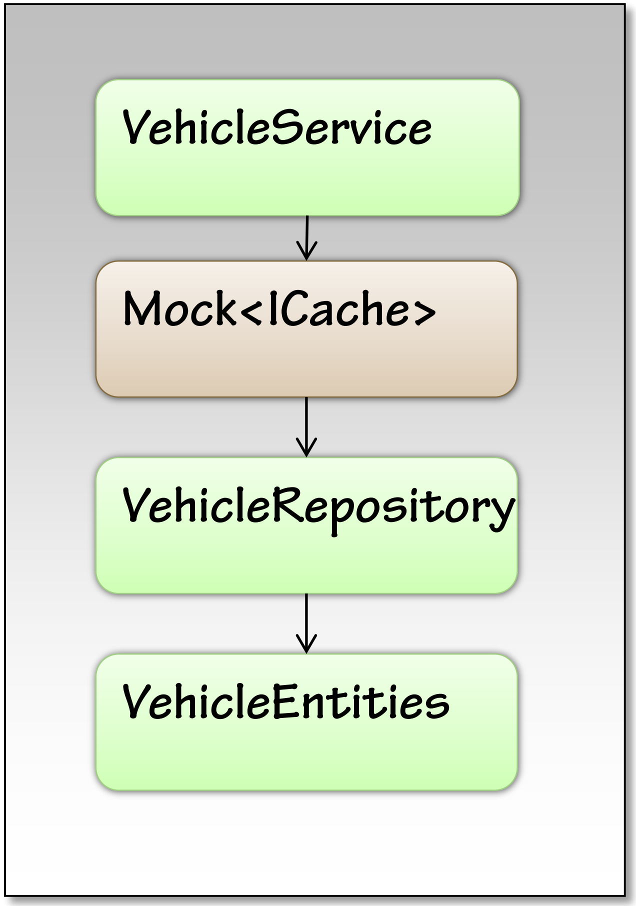
At Build Time

- **Unit tests**
  - □ Moq
  - □ Mock ICache – test workflow



At Build Time

- **Unit tests**
  - □ Moq
  - □ Mock ICache – test workflow
  - □ Verify get/check/set pattern


VehicleService

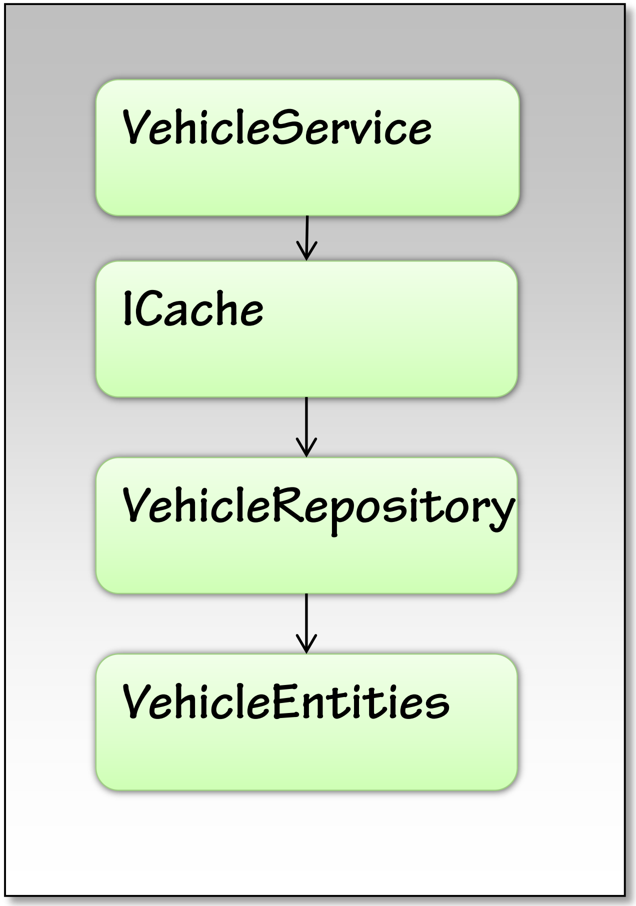
Mock<ICache>

VehicleRepository

VehicleEntities

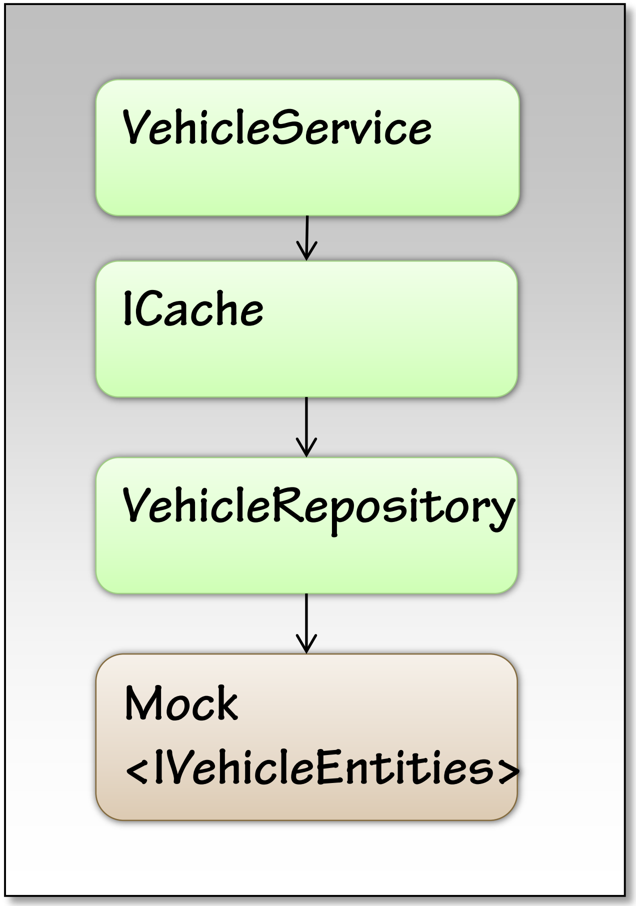
At Build Time

- **Unit tests**
  - Moq
  - Mock dependencies - test cache



At Build Time

- **Unit tests**
  - Moq
  - Mock dependencies - test cache
  - Verify business logic
  - Verify cache stack
    - Enabled=false



At Build Time

- Demo: Integration Tests

At Build Time

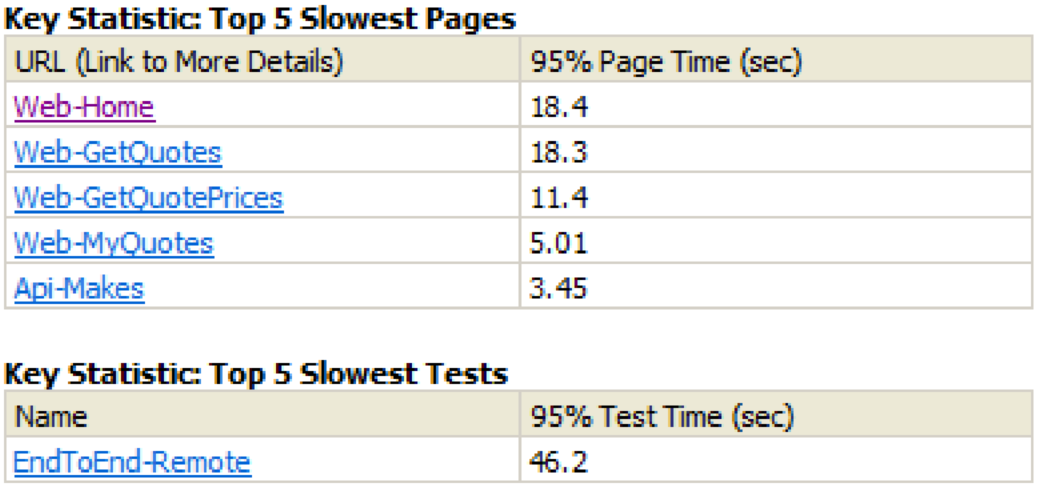
- **Integration web tests**
  - Feature-level testing
  - Include dependencies
    - Database
    - Downstream services
    - Cache stores
- **Verify features**
  - Response time goal
  - Response cache headers
  - Correctness, with & without caching

At Build Time

- Demo: Performance Tests

At Build Time

- **Performance tests**
  - Use end-to-end web test
  - Load patterns, stress test
- **Before caching: 60 concurrent users**
  - Page load: 5-18 seconds
  - Scenario: 46 seconds



Key Statistic: Top 5 Slowest Pages

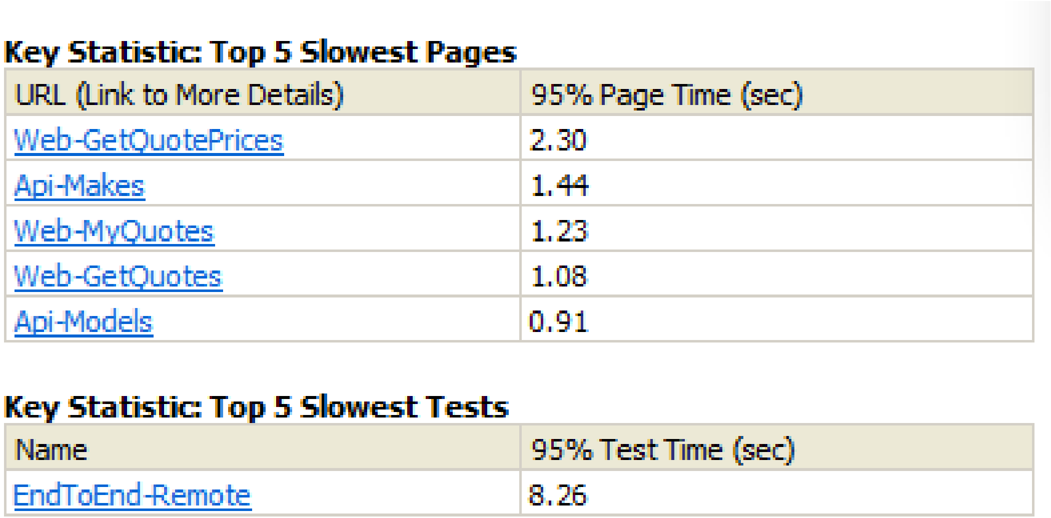
URL (Link to More Details)	95% Page Time (sec)
Web-Home	18.4
Web-GetQuotes	18.3
Web-GetQuotePrices	11.4
Web-MyQuotes	5.01
Api-Makes	3.45

Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
EndToEnd-Remote	46.2

At Build Time

- **With caching: 60 concurrent users**
  - Page load: 1-2 seconds
  - Scenario: 8 seconds



Key Statistic: Top 5 Slowest Pages

URL (Link to More Details)	95% Page Time (sec)
Web-GetQuotePrices	2.30
Api-Makes	1.44
Web-MyQuotes	1.23
Web-GetQuotes	1.08
Api-Models	0.91

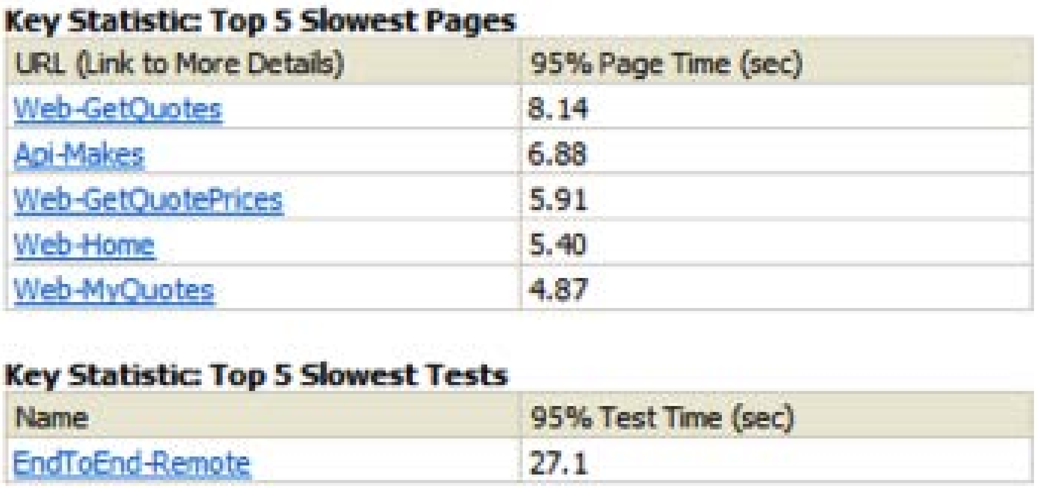
Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
EndToEnd-Remote	8.26

  - Same infrastructure, same functionality, same load
  - End-to-end 8 seconds, from 46 seconds
  - Slowest page 2.3 seconds, from 18 seconds

At Build Time

- **With caching: 120 concurrent users**
  - Page load: 5-8 seconds
  - Scenario: 27 seconds



Key Statistic: Top 5 Slowest Pages

URL (Link to More Details)	95% Page Time (sec)
Web-GetQuotes	8.14
Api-Makes	6.88
Web-GetQuotePrices	5.91
Web-Home	5.40
Web-MyQuotes	4.87

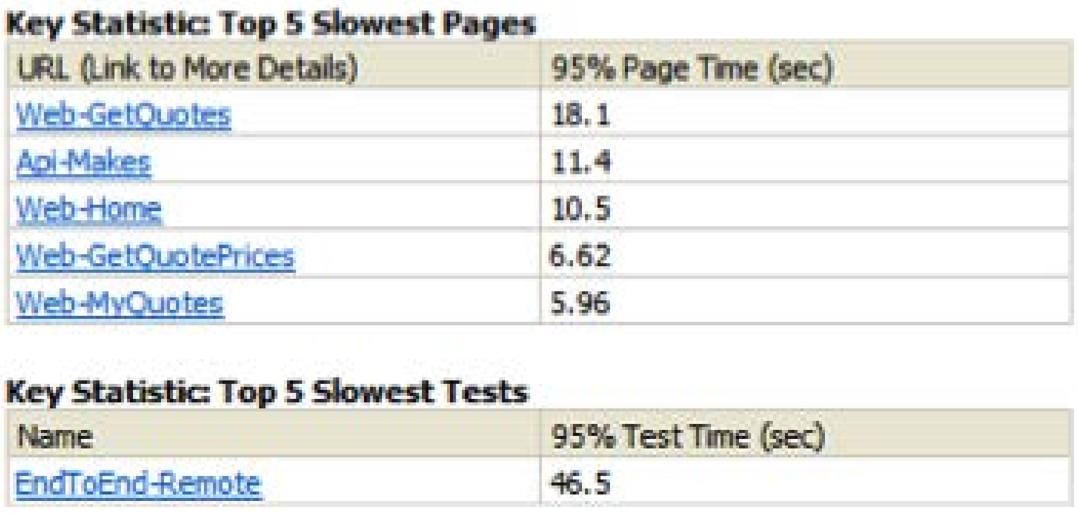
Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
EndToEnd-Remote	27.1

  - Same infrastructure, same functionality, **double** load
  - End-to-end 27 seconds, from 46 seconds
  - Slowest page 8 seconds, from 18 seconds

At Build Time

- **With caching: 180 concurrent users**
  - Page load: 5-18 seconds
  - Scenario: 46 seconds



Key Statistic: Top 5 Slowest Pages

URL (Link to More Details)	95% Page Time (sec)
Web-GetQuotes	18.1
Api-Makes	11.4
Web-Home	10.5
Web-GetQuotePrices	6.62
Web-MyQuotes	5.96

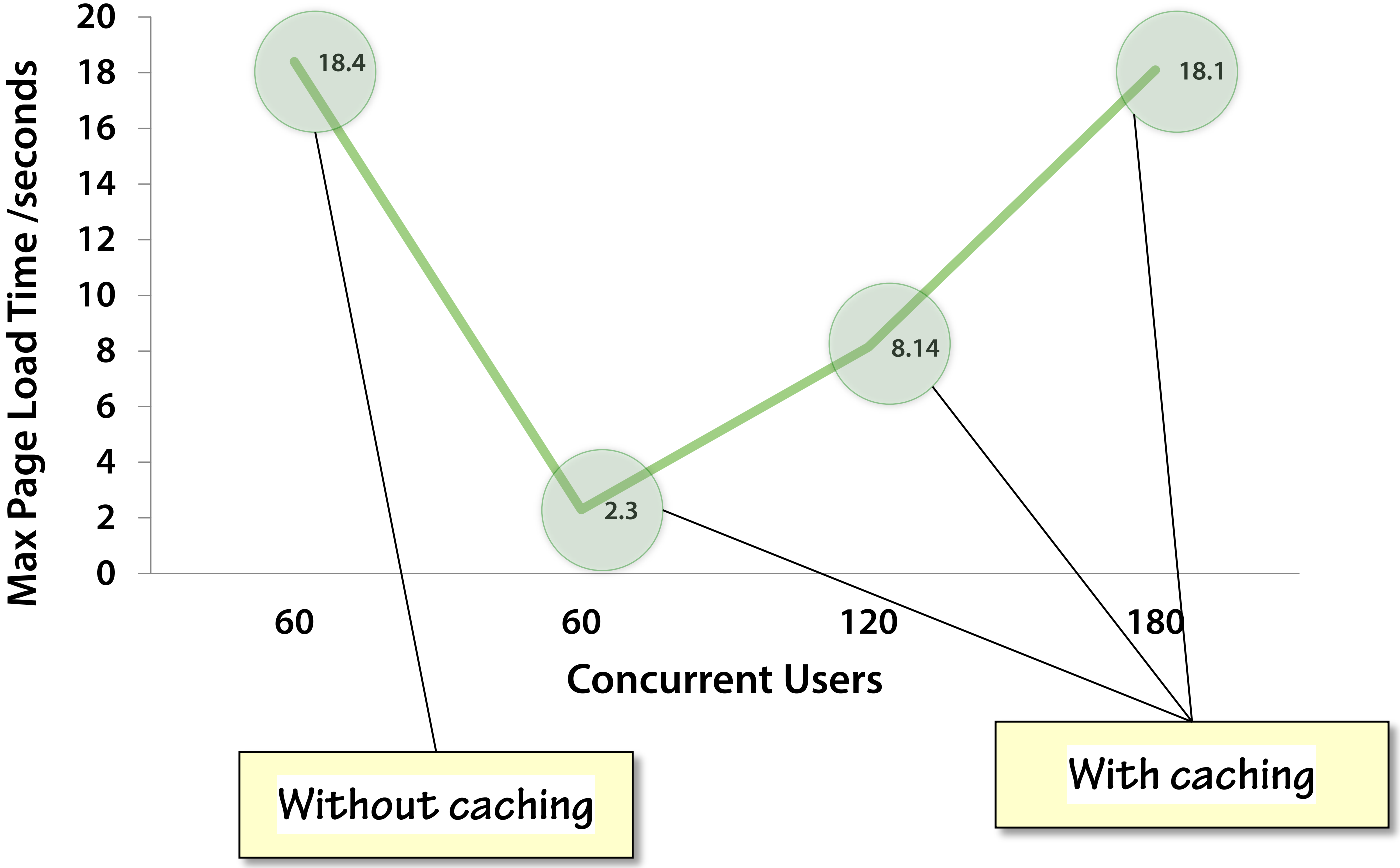
Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
EndToEnd-Remote	46.5

  - Same infrastructure, same functionality, **treble** load
  - Same performance

At Build Time

- **Headline stats**



At Run Time

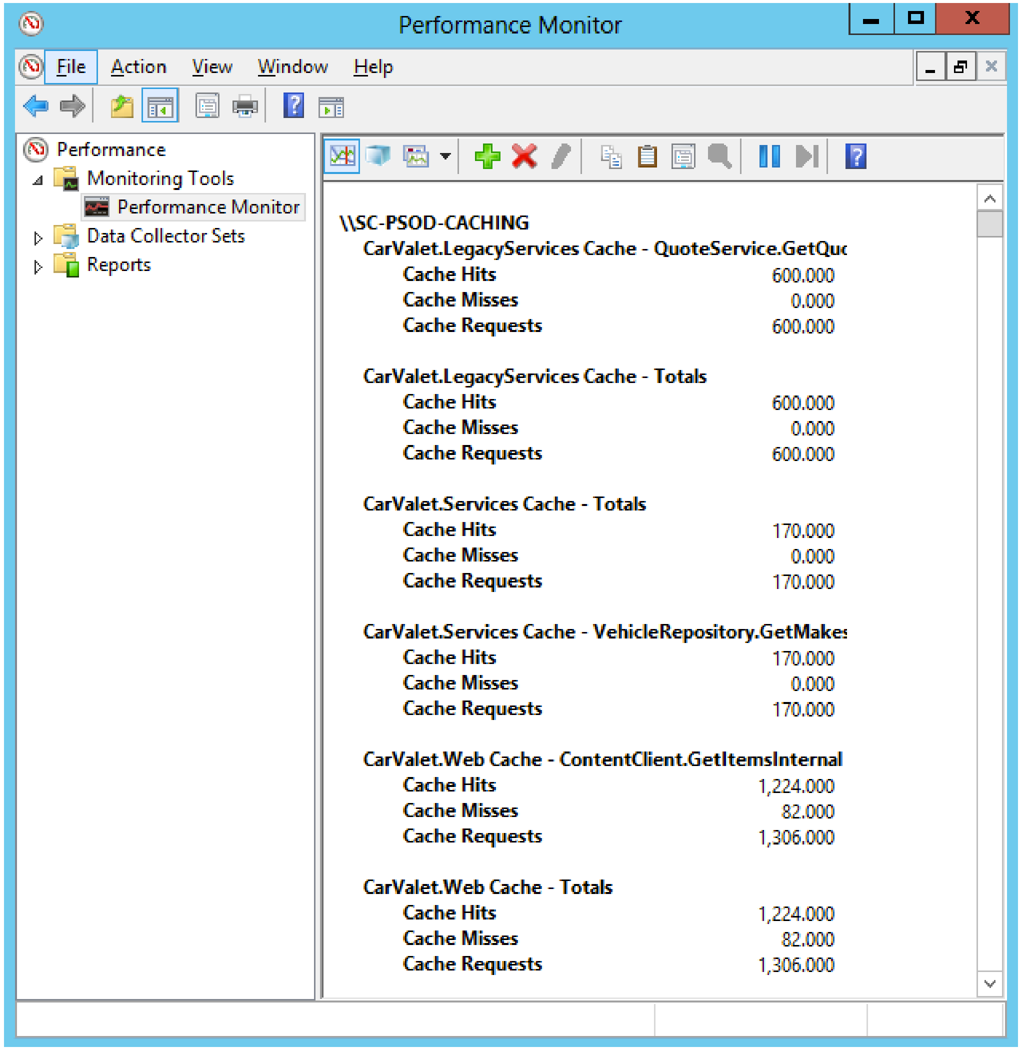
- **Performance counters**
- **Cache store counters**
  - Overall stats
- **Custom cache counters**
  - Fine-grained cache statistics
  - Cost of initialisation
  - Usage cost

At Run Time

- Demo

At Run Time

- **Custom cache counters**
  - Requests, hits & misses
  - Total per component
  - Total per cache target



```
\\SC-PSOD-CACHING
  CarValet.LegacyServices Cache - QuoteService.GetQuc
      Cache Hits                                    600.000
      Cache Misses                                    0.000
      Cache Requests                                600.000

  CarValet.LegacyServices Cache - Totals
      Cache Hits                                    600.000
      Cache Misses                                    0.000
      Cache Requests                                600.000

  CarValet.Services Cache - Totals
      Cache Hits                                    170.000
      Cache Misses                                    0.000
      Cache Requests                                170.000

  CarValet.Services Cache - VehicleRepository.GetMakes
      Cache Hits                                    170.000
      Cache Misses                                    0.000
      Cache Requests                                170.000

  CarValet.Web Cache - ContentClient.GetItemsInternal
      Cache Hits                                  1,224.000
      Cache Misses                                   82.000
      Cache Requests                              1,306.000

  CarValet.Web Cache - Totals
      Cache Hits                                  1,224.000
      Cache Misses                                   82.000
      Cache Requests                              1,306.000
```

At Run Time

- **Sample results – primed cache, stepped load**

```
CarValet.LegacyServices Cache - QuoteService.GetQuotesInternal
Cache Hits            600
Cache Misses            0
Cache Requests        600

CarValet.Services Cache - VehicleRepository.GetMakes
Cache Hits            170
Cache Misses            0
Cache Requests        170

CarValet.Web Cache - ContentClient.GetItemsInternal
Cache Hits           1224
Cache Misses           82
Cache Requests       1306
```

Summary

- **Automated testing**
  - Unit tests
  - Integration tests
- **Performance tests**
  - Baseline stats
- **Instrumentation**
  - Performance counters
  - Component monitoring