Performance

Nick Sumner
Performance

- Performance is a key nonfunctional criterion, but it is often poorly analyzed.
Performance

- Performance is a key nonfunctional criterion, but it is often poorly analyzed.

- Need to keep several several issues in mind:
Performance

- Performance is a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several issues in mind:
  - What facets of performance matter?
Performance

- Performance is a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several several issues in mind:
  - What facets of performance matter?
  - What component/granularity matters?
Performance

- Performance is a key nonfunctional criterion, but it is often poorly analyzed.

- Need to keep several several issues in mind:
  - What facets of performance matter?
  - What component/granularity matters?
  - How can we appropriately measure/ensure them?
Performance

- Performance is a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several issues in mind:
  - What facets of performance matter?
  - What component/granularity matters?
  - How can we appropriately measure/ensure them?
  - How should we report/present the results?
Performance

- Performance is a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several several issues in mind:
  - What facets of performance matter?
  - What component/granularity matters?
  - How can we appropriately measure/ensure them?
  - How should we report/present the results?
  - How can this fit into the overall QA process?
Facets of Performance

- What do we mean by performance?
Facets of Performance

- What do we mean by performance?
  - **Speed/Runtime**
    - Total running time required
Facets of Performance

- What do we mean by performance?
  - **Speed/Run Time**
    - Total running time required
  - **Throughput**
    - Pages/Transactions per second, bytes per second
Facets of Performance

• What do we mean by performance?
  – **Speed/Runtime**
    • Total running time required
  – **Throughput**
    • Pages/Transactions per second, bytes per second
  – **Responsiveness**
    • UI response time, server response time at peak load
Facets of Performance

- What do we mean by performance?
  - **Speed/Runtime**
    - Total running time required
  - **Throughput**
    - Pages/Transactions per second, bytes per second
  - **Responsiveness**
    - UI response time, server response time at peak load
  - **Memory Consumption**
    - Peak memory consumption
  - ...
Facets of Performance

- All of these (and more) are valid concerns
Facets of Performance

- All of these (and more) are valid concerns
- You must think critically about what aspects are most important to
  - End users?
Facets of Performance

- All of these (and more) are valid concerns
- You must think critically about what aspects are most important to
  - End users?
  - Operators?
Coarseness of Performance

- Performance may matter at different levels
Coarseness of Performance

- Performance may matter at different levels
  - Whole system
Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level
Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level

- May be consider performance at many different levels, but they may not be independent!
Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level

- May be consider performance at many different levels, but they may not be independent!

How are the different levels related?
Coarseness of Performance

• Performance may matter at different levels
  – Whole system
  – Critical components
  – Instruction level architectural level

• May be consider performance at many different levels, but they may not be independent!
  – Whole system & critical components are monitored for performance.
Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level

- May be consider performance at many different levels, but they may not be independent!
  - Whole system & critical components are monitored for performance.
  - Instruction and architectural level issues are examined only on demand when other problems are noted.
Coarseness of Performance

- System level issues?
Coarseness of Performance

- System level issues?
  - Design
  - Algorithms
Coarseness of Performance

- System level issues?
  - Design
  - Algorithms
- Instruction/Architecture level issues?
Coarseness of Performance

- System level issues?
  - Design
  - Algorithms

- Instruction/Architecture level issues?
  - Cache misses
  - Branch misprediction
  - Underutilization of resources (SIMD, multicore, ...)

**Coarseness of Performance**

- **System level issues?**
  - Design
  - Algorithms

- **Instruction/Architecture level issues?**
  - Cache misses
  - Branch misprediction
  - Underutilization of resources (SIMD, multicore, ...)

- **Use profilers to gain information at a certain level**
  - perf, gprof, cachegrind, oprofile, java mission control
  - Purify, VS Profiling Tools
  - Coz, ...
So how can we measure these?

- Idea: Run the test suite and see how long it takes
Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes

  Is this reasonable? Why/why not?
Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes
    - How well does this capture system level performance?
    - ““ low level performance?
Measurement

• So how can we measure these?
  – Idea: Run the test suite and see how long it takes
    • How well does this capture system level performance?
    • “ low level performance?

• A *functionality* based test suite will not reflect the *performance* concerns!
Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes
    - How well does this capture system level performance?
    - “low level performance?”
- A *functionality* based test suite will not reflect the *performance* concerns!
  - Need to design a new set of tests specifically targeting performance issues.
Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes
    - How well does this capture system level performance?
    - “ low level performance?
- A *functionality* based test suite will not reflect the *performance* concerns!
  - Need to design a new set of tests specifically targeting performance issues.

How? What should the tests capture?
Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
  - Hot paths, tight loops, frequently called functions, ...
Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
  - Hot paths, tight loops, frequently called functions, ...

- Step 2: Go back to step 1 and make sure it is realistic
  - People commonly use synthetic workloads for performance measurement and are misled by results.
Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
  - Hot paths, tight loops, frequently called functions, ...

- Step 2: Go back to step 1 and make sure it is realistic
  - People commonly use synthetic workloads for performance measurement and are misled by results.

Why is this an issue?
**Measurement**

- **Step 1:** Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
  - Hot paths, tight loops, frequently called functions, ...

- **Step 2:** Go back to step 1 and make sure it is realistic
  - People commonly use synthetic workloads for performance measurement and are misled by results.
    - Too simple
    - Doesn't reflect interaction complexities of real code
    - ...

Measurement

- Step 3: Run & measure your workload
Measurement

- Step 3: Run & measure your workload
  - Is this good enough?

```java
startTime = getCurrentTimeInSeconds();
doWorkloadOfInterest();
endTime = getCurrentTimeInSeconds();
reportResult(endTime - startTime);
```
Measurement

- Step 3: Run & measure your workload
  - Is this good enough?

```java
startTime = getCurrentTimeInSeconds();
doWorkloadOfInterest();
endTime = getCurrentTimeInSeconds();
reportResult(endTime - startTime);
```

- What can go wrong?
Measurement

- Step 3: Run & measure your workload
  - Is this good enough?

```java
startTime = getCurrentTimeInSeconds();
doWorkloadOfInterest();
endTime = getCurrentTimeInSeconds();
reportResult(endTime - startTime);
```

- What can go wrong?
  - Timing granularity & units
  - Warm up time
  - Nondeterministic behavior
  - ...
Measurement

- Granularity & Units
  - Why is granularity a problem?
Measurement

- Granularity & Units
  - Why is granularity a problem?
  - What are alternatives to getCurrentTimeInSeconds?
Measurement

- Granularity & Units
  - Why is granularity a problem?
  - What are alternatives to getCurrentTimeInSeconds?
  - What if I want to predict performance on a different machine?
Measurement

• Granularity & Units
  – Why is granularity a problem?
  – What are alternatives to getCurrentTimeInSeconds?
  – What if I want to predict performance on a different machine?

  • Using cycles instead of wall clock time can be useful, but has its own limitations
Measurement

- Warm up time
  - Why is warm up time necessary in general?
Measurement

- Warm up time
  - Why is warm up time necessary in general?
  - Why is it especially problematic for systems like Java?
Measurement

- Warm up time
  - Why is warm up time necessary \textit{in general}?
  - Why is it especially problematic for systems like Java?
  - How can we modify our example to facilitate this?
Measurement

- Warm up time
  - Why is warm up time necessary *in general*?
  - Why is it especially problematic for systems like Java?
  - How can we modify our example to facilitate this?

```java
for (...) doWorkloadOfInterest();
startTime = getCurrentTimeInSeconds();
doWorkloadOfInterest();
endTime = getCurrentTimeInSeconds();
reportResult(endTime - startTime);
```
Measurement

• Nondeterministic behavior
  – Will `getCurrentTimeInSeconds();` always return the same number?

  Why/why not?
Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?
  - So what reflects a meaningful result?
    - Hint: *The Law of Large Numbers!*
Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?
  - So what reflects a meaningful result?
    - Hint: The Law of Large Numbers!

- By running the same test many times, the arithmetic mean will converge on the expected value
Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?
  - So what reflects a meaningful result?
    - Hint: The Law of Large Numbers!

- By running the same test many times, the arithmetic mean will converge on the expected value
  - So have we solved the problem?
Measurement

* Nondeterministic behavior
  - Will getCurrentTimeInSeconds(); always return the same number?
  - So what reflects a meaningful result?
    * Hint: The Law of Large Numbers!
  
* By running the same test many times, the arithmetic mean will converge on the expected value
  - So have we solved the problem?
  - When do we stop?
Measurement

• Nondeterministic behavior
  – Will `getCurrentTimeInSeconds();` always return the same number?
  – So what reflects a meaningful result?
    • Hint: The Law of Large Numbers!
• By running the same test many times, the arithmetic mean will converge on the expected value
  – So have we solved the problem?
  – When do we stop?

There's a bit more to deal with here. We'll revisit it.
Measurement

- A revised (informal) approach:

```java
for (...) doWorkloadOfInterest();
startTime = getCurrentTimeInNanos();
for (...) doWorkloadOfInterest();
endTime = getCurrentTimeInNanos();
reportResult(endTime - startTime);
```
Congratulations! You have measured performance results of your application.

Test1 → Result1
Test2 → Result2
Test3 → Result4
Test5 → Result5
Test6 → Result6
Test7 → Result7

Are the results good or bad? How do you know?
Reporting

Congratulations! You have measured performance results of your application.

Test1  ➔  Result1
Test2  ➔  Result2
Test3  ➔  Result4
Test5  ➔  Result5
Test6  ➔  Result6
Test7  ➔  Result7

Are the results good or bad? How do you know?

The interpretation of results is only meaningful given context and appropriate presentation.
Reporting

- Often, performance is tracked across versions

<table>
<thead>
<tr>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result1</td>
<td>Result1</td>
</tr>
<tr>
<td>Result2</td>
<td>Result2</td>
</tr>
<tr>
<td>Result4</td>
<td>Result4</td>
</tr>
<tr>
<td>Result5</td>
<td>Result5</td>
</tr>
<tr>
<td>Result6</td>
<td>Result6</td>
</tr>
<tr>
<td>Result7</td>
<td>Result7</td>
</tr>
</tbody>
</table>
### Reporting

- Often, performance is tracked across versions

<table>
<thead>
<tr>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result1</td>
<td>Result1</td>
</tr>
<tr>
<td>Result2</td>
<td>Result2</td>
</tr>
<tr>
<td>Result4</td>
<td>Result4</td>
</tr>
<tr>
<td>Result5</td>
<td>Result5</td>
</tr>
<tr>
<td>Result6</td>
<td>Result6</td>
</tr>
<tr>
<td>Result7</td>
<td>Result7</td>
</tr>
</tbody>
</table>

- Given this, how can we tell if there is an **improvement** or a **problem**?
Reporting

Strategies

- Sample Histograms
Reporting

Strategies

• Sample Histograms

- Old
- New
Reporting

Strategies

• Sample Histograms

  - Both gains & losses can be noted
Reporting

Strategies

- Sample Histograms

- Both gains & losses can be noted

- Distribution of changes
Reporting

Strategies

- Sample Histograms
  - Both gains & losses can be noted

- Distribution of changes
Reporting

Strategies

• Sample Histograms
  – Both gains & losses can be noted

• Distribution of changes
Reporting

- You might have to summarize the changes using a single number.
You might have to summarize the changes using a single number

- Median?
Reporting

- You might have to summarize the changes using a single number
  - Median?
Reporting

- You might have to summarize the changes using a single number
  - Median?
  - Mode?
You might have to summarize the changes using a single number

- Median?
- Mode?
You might have to summarize the changes using a single number

- Median?
- Mode?
- Mean?
You might have to summarize the changes using a single number

- Median?
- Mode?
- Mean?
Reporting

- You might have to summarize the changes using a single number
  - Median?
  - Mode?
  - Mean?

All of them can be misleading in some way! Yet you'll still likely be expected to use them.
You might have to summarize the changes using a single number

- Median?
- Mode?
- Mean?

But what do we even mean by... mean/average?
You might have to summarize the changes using a single number

- Median?
- Mode?
- Mean?

But what do we even mean by... mean/average?

- There are actually many types of averages
Reporting

Averages of $r_1, r_2, \ldots, r_N$

- Many ways to measure expectation or tendency
Averages of $r_1, r_2, \ldots, r_N$

- Many ways to measure *expectation* or *tendency*
- Arithmetic Mean

$$\frac{1}{N} \sum_{i=1}^{N} r_i$$
Averages of $r_1, r_2, \ldots, r_N$

- Many ways to measure *expectation* or *tendency*
- Arithmetic Mean
  \[
  \frac{1}{N} \sum_{i=1}^{N} r_i
  \]
- Harmonic Mean
  \[
  \frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
  \]
Averages of \( r_1, r_2, \ldots, r_N \)

- Many ways to measure expectation or tendency
- Arithmetic Mean
  \[
  \frac{1}{N} \sum_{i=1}^{N} r_i
  \]
- Harmonic Mean
  \[
  \frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
  \]
- Geometric Mean
  \[
  \sqrt[N]{\prod_{i=1}^{N} r_i}
  \]
Averages of $r_1, r_2, ..., r_N$

- Many ways to measure *expectation* or *tendency*
- Arithmetic Mean
  \[
  \frac{1}{N} \sum_{i=1}^{N} r_i
  \]
- Harmonic Mean
  \[
  \frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
  \]
- Geometric Mean
  \[
  \sqrt[N]{\prod_{i=1}^{N} r_i}
  \]

Each type means something different and has valid uses
Reporting

- Arithmetic Mean
  - Good for reporting averages of numbers that mean the same thing

\[ \frac{1}{N} \sum_{i=1}^{N} r_i \]
Reporting

- **Arithmetic Mean**
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing *sample means*

\[ \frac{1}{N} \sum_{i=1}^{N} r_i \]
Reporting

- **Arithmetic Mean**
  \[
  \frac{1}{N} \sum_{i=1}^{N} r_i
  \]
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times
Reporting

- **Arithmetic Mean**
  \[
  \frac{1}{N} \sum_{i=1}^{N} r_i
  \]
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times

Handling Nondeterminism

```plaintext
for (x in 0 to 4)
    times[x] = doWorkloadOfInterest();

E(time) = arithmean(times)
```
Reporting

- **Arithmetic Mean**
  \[
  \frac{1}{N} \sum_{i=1}^{N} r_i
  \]
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times

- **Harmonic Mean**
  \[
  \frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
  \]
  - Good for reporting rates
Reporting

- Arithmetic Mean
  \[ \frac{1}{N} \sum_{i=1}^{N} r_i \]
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times

- Harmonic Mean
  \[ \frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}} \]
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks
Reporting

Given tasks t1, t2, & t3 serving 40 pages each:

\[
\begin{align*}
\text{throughput(t1)} &= 10 \text{ pages/sec} \\
\text{throughput(t2)} &= 20 \text{ pages/sec} \\
\text{throughput(t3)} &= 20 \text{ pages/sec}
\end{align*}
\]

What is the average throughput? What should it mean?

- **Harmonic Mean**
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

\[
\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
\]
Reporting

Given tasks $t_1$, $t_2$, & $t_3$ serving 40 pages each:

- $\text{throughput}(t_1) = 10 \text{ pages/sec}$
- $\text{throughput}(t_2) = 20 \text{ pages/sec}$
- $\text{throughput}(t_3) = 20 \text{ pages/sec}$

What is the average throughput? What should it mean?
Arithmetic $= 16.7 \text{ p/s}$

- **Harmonic Mean**
  
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

$$
\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
$$
Reporting

- **Arithmetic Mean**
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times

- **Harmonic Mean**
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

Given tasks t1, t2, & t3 serving 40 pages each:
  - throughput(t1) = 10 pages/sec
  - throughput(t2) = 20 pages/sec
  - throughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?
Arithmetic = 16.7 p/s       Harmonic = 15 p/s

\[
\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
\]
**Reporting**

<table>
<thead>
<tr>
<th>Arithmetic Mean</th>
<th>Harmonic Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good for reporting averages of numbers that mean the same thing</td>
<td>Good for reporting rates</td>
</tr>
<tr>
<td>Used for computing sample means</td>
<td>e.g. Required throughput for a set of tasks</td>
</tr>
</tbody>
</table>

Given tasks t1, t2, & t3 serving 40 pages each:
- throughput(t1) = 10 pages/sec
- throughput(t2) = 20 pages/sec
- throughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?
- Arithmetic = 16.7 p/s
- Harmonic = 15 p/s

\[
\frac{120}{16.7} = 7.2 \quad \frac{120}{15} = 8
\]

- Harmonic Mean
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

\[
\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
\]
### Reporting

Given tasks t1, t2, & t3 serving 40 pages each:
- throughput(t1) = 10 pages/sec
- throughput(t2) = 20 pages/sec
- throughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?

- **Arithmetic** = \(\frac{120}{16.7} = 7.2\) p/s
- **Harmonic** = \(\frac{120}{15} = 8\) p/s

### Harmonic Mean

- Good for reporting rates
- e.g. Required throughput

\[-\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}\]

Identifies the constant rate required for the same time.
Reporting

Given tasks t1, t2, & t3 serving 40 pages each:

\[
\text{throughput(t1)} = 10 \text{ pages/sec} \\
\text{throughput(t2)} = 20 \text{ pages/sec} \\
\text{throughput(t3)} = 20 \text{ pages/sec}
\]

What is the average throughput? What should it mean?

Arithmetic = 16.7 p/s  
Harmonic = 15 p/s

\[
\frac{120}{16.7} = 7.2 \quad \frac{120}{15} = 8
\]

- Harmonic Mean
  - Good for reporting rates
  - e.g. Required throughput

\[
\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}
\]

Identifies the constant rate required for the same time

CAVEAT: If the size of each workload changes, a weighted harmonic mean is required!
Reporting

- Geometric Mean
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks
Reporting

• Geometric Mean
  – Good for reporting results that mean different things
  – e.g. Timing results across many different benchmarks

\[ N \sqrt[N]{\prod_{i=1}^{N} r_i} \]

Any idea why it may be useful here? (A bit of a thought experiment)
Reporting

- Geometric Mean
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks

\[
\sqrt[N]{\prod_{i=1}^{N} r_i}
\]
Reporting

• Geometric Mean
  – Good for reporting results that mean different things
  – e.g. Timing results across *many different* benchmarks

\[ N \sqrt[ N ]{ \prod_{ i=1 }^{ N } r_i } \]

What happens to the arithmetic mean?

\{ halved \}

Old

T1
T2

New 1

T1
T2
• Geometric Mean
  – Good for reporting results that mean different things
  – e.g. Timing results across many different benchmarks

\[ N \sqrt[N]{\prod_{i=1}^{N} r_i} \]

What happens to the arithmetic mean?
Reporting

- Geometric Mean
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks

\[ N \sqrt[N]{ \prod_{i=1}^{N} r_i } \]

The (non) change to T1 dominates any behavior for T2!
Reporting

- Geometric Mean
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks

\[
\sqrt[N]{\prod_{i=1}^{N} r_i}
\]

Geometric:

\[
\sqrt{r_1 \times r_2}
\]
Reporting

- Geometric Mean
  - Good for reporting results that mean different things
  - e.g. Timing results across many different benchmarks

\[ \sqrt[N]{\prod_{i=1}^{N} r_i} \]

Geometric:

Old

\[ \sqrt{r_1 \times r_2} \]

Old

\[ \sqrt{r_1 \times \left( \frac{1}{2} r_2 \right)} \]

New 1
Reporting

- Geometric Mean
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks

\[
N \sqrt[\prod_{i=1}^{N} r_i]{}
\]

<table>
<thead>
<tr>
<th></th>
<th>Old</th>
<th>Old</th>
<th>Old</th>
<th>New 1</th>
<th>New 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td></td>
<td>(\sqrt{r_1 \times r_2})</td>
<td>(r_1 \times \left(\frac{1}{2} r_2\right))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reporting

- Geometric Mean
  - Good for reporting results that mean different things
  - e.g. Timing results across many different benchmarks

Geometric:

\[ \sqrt[N]{\prod_{i=1}^{N} r_i} \]

Old

\[ \sqrt{r_1 \times r_2} \]

\[ \sqrt{r_1 \times \left(\frac{1}{2} r_2\right)} \]

\[ \sqrt{\frac{1}{2} \times r_1 \times r_2} = \sqrt{\left(\frac{1}{2} r_1\right) \times r_2} \]

Old

New 1

New 2
Reporting

• Geometric Mean
  – Good for reporting results that mean different things
  – e.g. Timing results across many different benchmarks
  – A 10% difference in any benchmark affects the final value the same way
Reporting

- Geometric Mean
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks
  - A 10% difference in any benchmark affects the final value the same way

Note: It doesn't have an intuitive meaning! It merely provides a balanced *score* of performance.
Reporting

- Nondeterminism
  - But all of these assume that the numbers across benchmarks are valid/deterministic
Reporting

• Nondeterminism
  – But all of these assume that the numbers across benchmarks are valid/deterministic
  – What does nondeterminism do?
Reporting

- Nondeterminism
  - But all of these assume that the numbers across benchmarks are valid/deterministic
  - What does nondeterminism do?
  - Every measurement is a sample from a probability distribution
Do these mean the same thing?
• Do these mean the same thing?

You should also worry about the *sample standard deviation*

\[
\sqrt{\frac{\sum (r_1 - \text{arithmean}(r))^2}{N}}
\]
Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If the standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design
Reporting

- **Standard Deviation**
  - Provides a notion of *confidence* in the sample mean
  - If the standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design

- **More rigorously, you can even compute**
  - Confidence intervals
Reporting

• Standard Deviation
  – Provides a notion of confidence in the sample mean
  – If there standard deviation is large:
    • You may need more samples to build confidence
    • There may be methodological error in your test design

• More rigorously, you can even compute
  – Confidence intervals
    • region in which actual value is likely to be
Reporting

- Standard Deviation
  - Provides a notion of confidence in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design

- More rigorously, you can even compute
  - Confidence intervals
  - Confidence tests (e.g. T-tests)
Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If the standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design

- More rigorously, you can even compute
  - Confidence intervals
  - Confidence tests (e.g. T-tests)
    - Test to determine whether the difference between sample means is ... meaningful
Reporting

- Standard Deviation
  - Provides a notion of confidence in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design

- More rigorously, you can even compute
  - Confidence intervals
  - Confidence tests (e.g. T-tests)
    To determine whether results are statistically significant
So what is often done in practice?

- Firefox(\url{http://arewefastyet.com/})
- LLVM (\url{http://llvm.org/perf/})