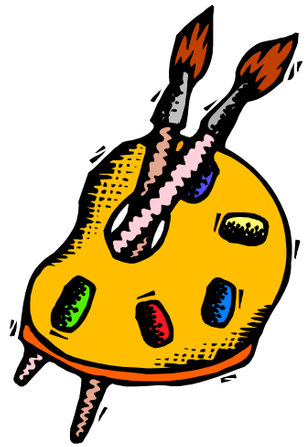
TPC – TC 2009

The Art of Building a Good Benchmark



Karl Huppler
TPC Chair
August 24, 2009

TPC Transaction Processing Performance Council

Seeming explosion of benchmarks over the last two decades

TPC-D

Storage Performance Council
Defining, administering, and promoting industry-standard, vendor-neutral benchmarks to characterize the performance of storage products

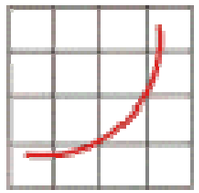
SYSmark2007

TPC-A

SPECjbb2005

SPECjms2007

TPC-B



SPECsfs2008

TPC-E

TPC-C

TPC-H

SPC-2C

TPC-R

TPC-App

SPECfp_rate2006

SPECjvm2008

SPECmail2008

BAPCO Real World, Real Benchmarks

SPECint_rate2000

SPECint2000

SPC-1C

SPECjms2007

SPECint2006

(and these are just a few!!!!)

TPC Transaction Processing Performance Council

Just because something is new,

doesn’t mean it is “good”

Successful Benchmark Requirements

Relevant

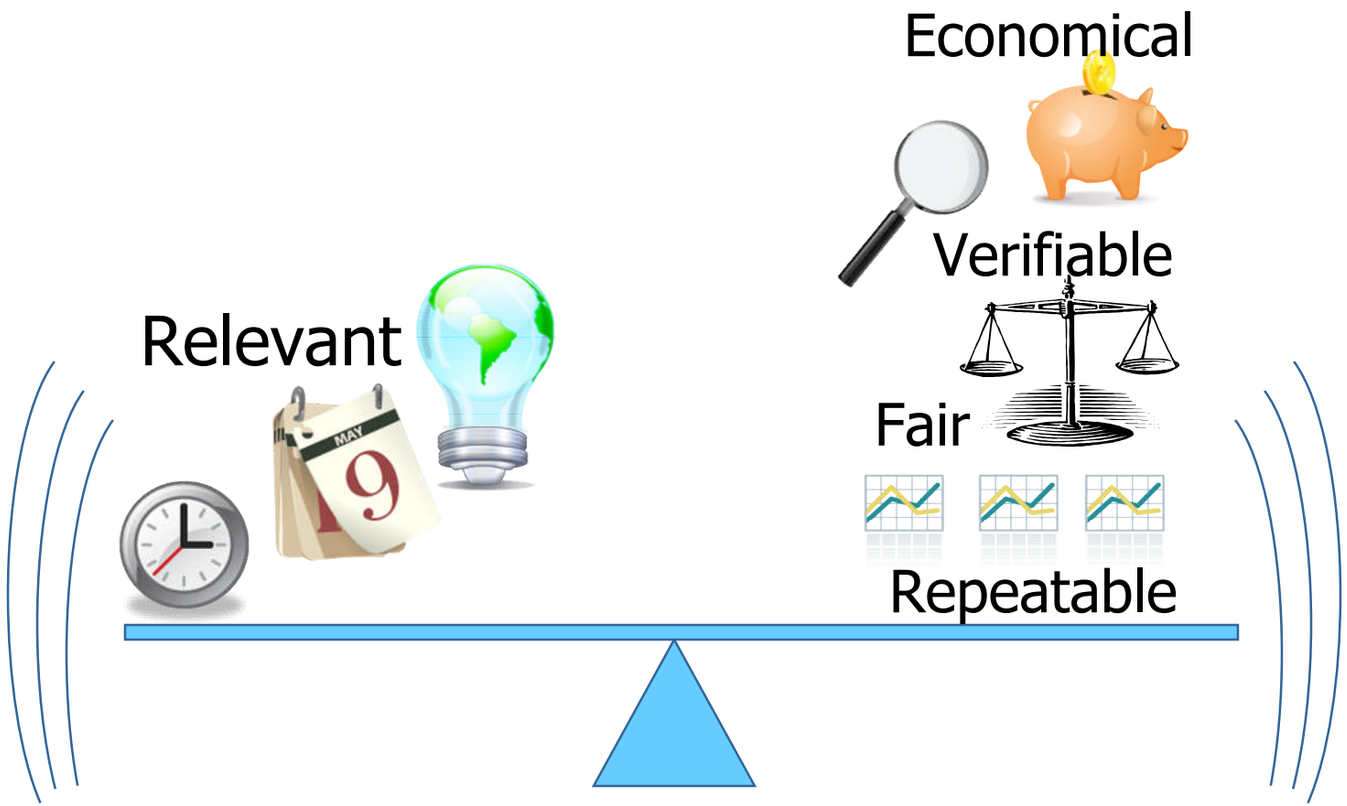
Repeatable

Fair

Verifiable

Economical

Compromise Required



TPC – TC 2009

Relevant Metric

1,223,457 SPECjbb bops

- Java business operations per second

4,801,497 tpmC

- transactions per minute in benchmark C

3,105 QphH@3000GB

$$\sqrt{\frac{3600 * SF}{\sqrt[24]{\prod_{i=1}^{i=22} QI(i,0) * \prod_{j=1}^{j=2} RI(j,0)}}} * [(S*22*3600)/Ts *SF]$$

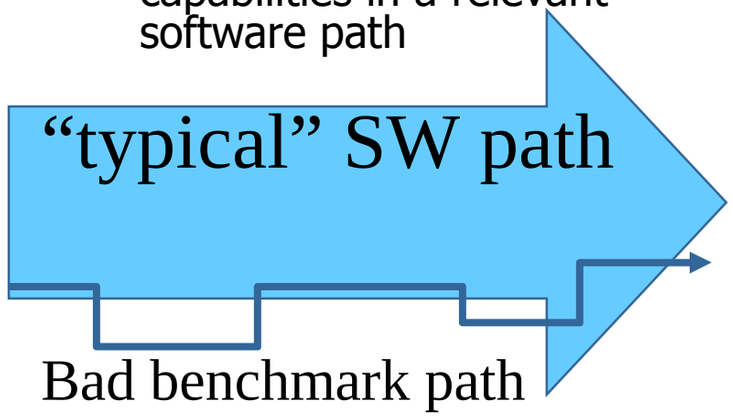
- (but it does look like queries per hour and it actually relates to that)

21.3 ?

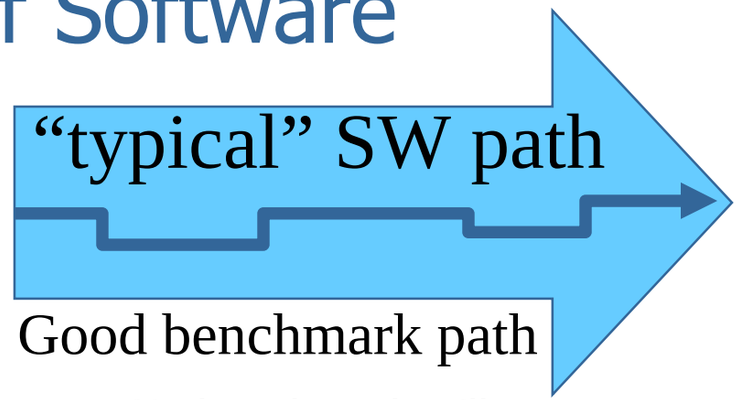
TPC Transaction Processing Performance Council

Use of Relevant of Software

- Use of appropriate software paths may be the most critical requirement of any benchmark
  - Encourages optimization of "real" consumer paths
  - Represents performance capabilities in a relevant software path



Bad benchmark path
– optimizes areas not important to consumer



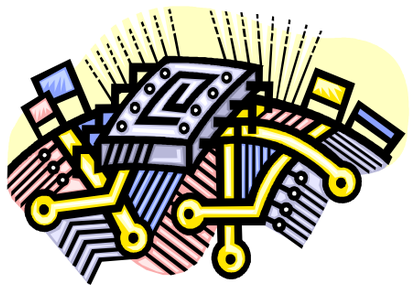
Good benchmark path

- No benchmark will exercise "all" important paths
- Good benchmarks run paths that are used by many applications in the business model of the benchmark
- Bad benchmarks use fringe paths whose optimization does not help real applications

TPC-C, TPC-H, SPECint, SPECfp, SPECjbb all are examples where benchmarks helped optimize consumer software paths.

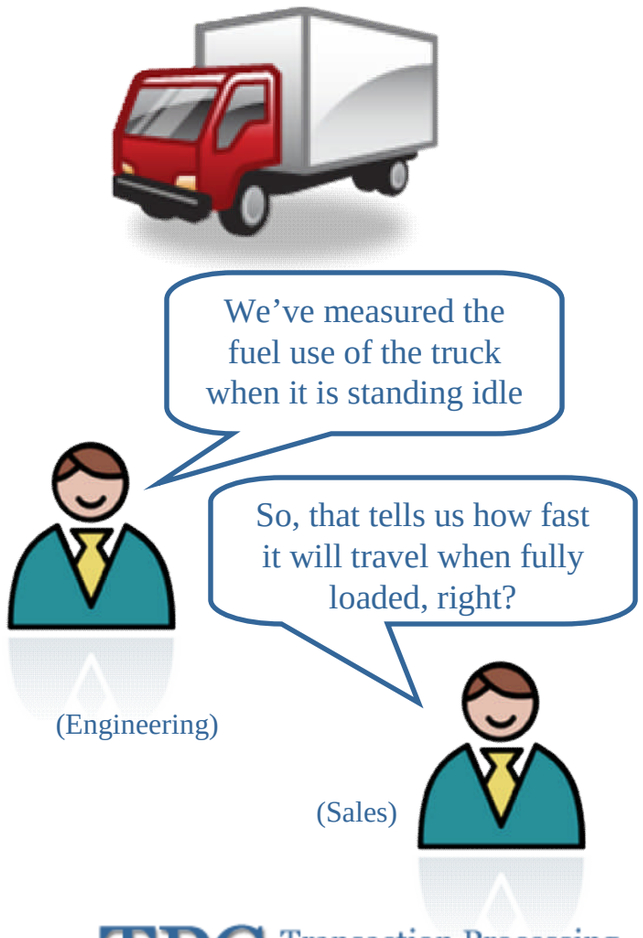
Use of Relevant Hardware

- Similar to software
  - Important that benchmark does not focus on hardware that is not important to "typical" applications in the business model of the benchmark
  - Wouldn't want a benchmark that only focused on a single, potentially low-use component, like a floating point accelerator
- Good benchmarks can help drive hardware design to eliminate consumer problems before they happen
  - At the macro level, TPC-C, after 17 years, continues to be used to provide system level engineering design guidance
  - Within the processor and related firmware, SPECcpu provides a broad range of stress points that can be used in early design and final proof points



Does what it says – no false claims

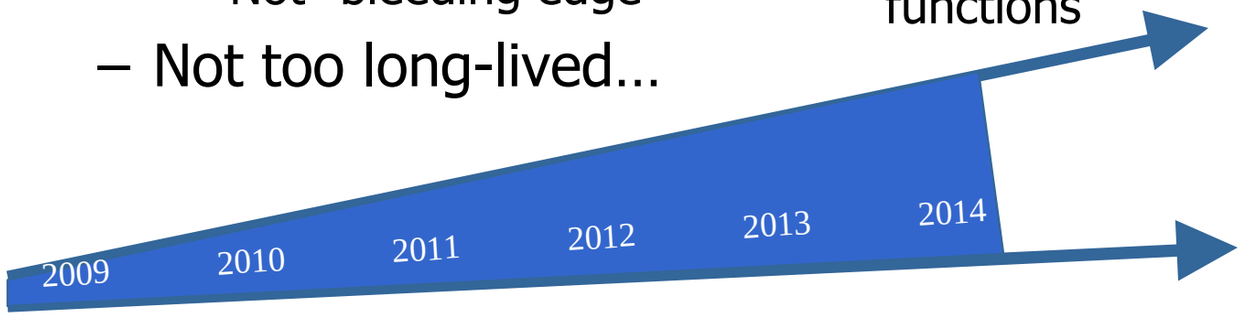
- Benchmarks must be designed to fit
  - A particular business model
  - A specific scope within the business model
- Benchmarks must clearly state that they cannot be generalized outside of the designed-for scope
- Sometimes, consumers of benchmark information make the wrong assumptions, anyway



TPC Transaction Processing Performance Council

Relevance – long life, broadly applicable

- Long-lived
  - Important functions
  - Current functions
    - Challenge: Not all product offerings will be at the same level
    - "leading edge"
    - Not "bleeding edge"
  - Not too long-lived…
- Broad Applicability
    - Business model may be tightly defined
    - Must not restrict applicability
  - TPC-C: General OLTP
  - SPECcpu: Broad suite of compute-intensive functions

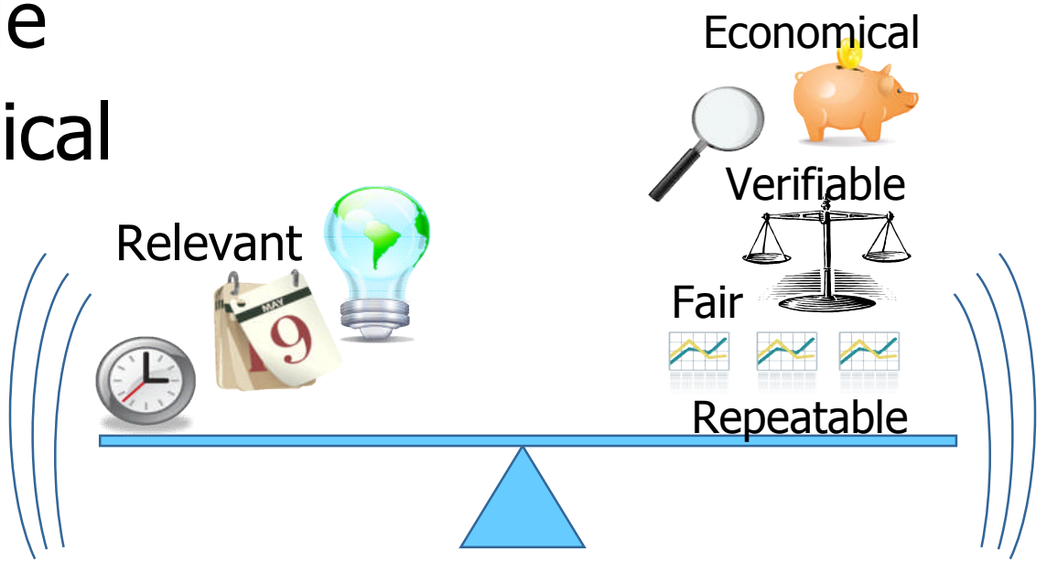


Strong Target Audience

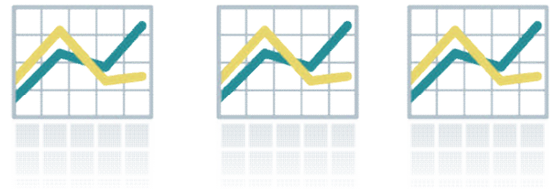
- After Software Relevance, most critical to building strong, long-lived, relevant benchmark
  - Not always known – new functions may not have current "audience"
  - Benchmarks "perfect in every way" may be retired because they cannot meet this
    - Can still be a good benchmark, if strong in many other areas
- Not always "consumers"
  - SPECcpu Audience:
    - Engineers, academics, designers, programmers (and sometimes marketing)
  - TPC-C Audience:
    - Engineers, designers, end-customers, analysts, marketing, executives, etc.

Trading Relevance for "Benchmarkability"

- Repeatable
- Fair and Portable
- Verifiable
- Economical



Repeatability



- Confidence in getting same result on each measurement
- Challenge when information changes
  - Queries run longer, different results
  - Disks respond differently, depending on prior use
- Compromises may be required
  - Pre-condition system by running application multiple times
    - Not real, but perhaps consistent
  - Ensure data changes do not affect future results
    - "sanitary" database where updates are in columns that are not queried, or inserts are with key values that are not queried
  - Refresh data on each run
    - (or as appropriate) – TPC-C lasts up to 12 hours

Fair and Portable

- Want benchmark to stress important, leading-edge features
  - Don't want to penalize strong solutions that have not optimized "all" of the new features
  - Do want to avoid reducing to functional "lowest common denominator"
- Focus on broad range of environments
  - … or declare that it is for a limited range
- Use of standard C, C++, Java, SQL makes portability easier
- Key requirement is testing on multiple platforms with multiple software environments
  - Ensures portability
  - Exposes inadvertent prejudice for the development environment
- Fair and Portable benchmarks trade custom and leading-edge features for broader applicability across environments

Verifiable

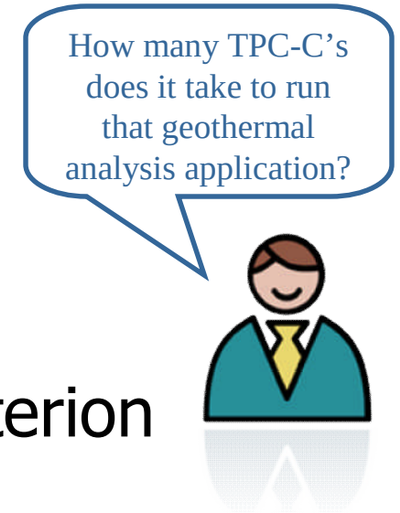
- Confidence in benchmark result required
- Can be self-verifying
  - Automatic routines built into benchmark to test against verification criteria
  - Many SPEC benchmarks do this, at least in part
- Can be reviewed and/or attested by a third party
  - TPC uses certified auditors who have demonstrated expertise in the benchmark
  - SPEC uses volunteer oversight of results from members of the development committee
  - Each method has advantages
- The easier the verification, the greater the confidence
  - May require trade-offs to simplify the benchmark

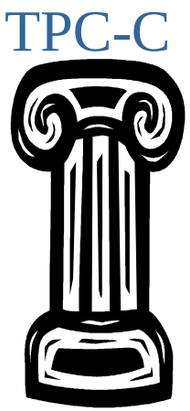
Economical

- An expensive benchmark requires great incentive to publish
  - Can still be a strong benchmark, but with a limited result set
- An inexpensive benchmark may become popular by sheer number-of-publishes
  - If coupled with the strength of other criteria in this discussion, can become very popular
  - SPECint2006, SPECfp2006, SPECint_rate2006, SPECfp_rate2006 are clear examples

Can't do it all

- A benchmark can be too "perfect"
  - Satisfying almost every criterion leads to
    - Too much popularity
    - Too wide a target audience
    - Too much difficulty to make changes
    - Too many general conclusions that are not based on the benchmark business model





Summary: Relating this to the TPC and future benchmark development

- The criteria in this discussion need to be kept in view throughout benchmark development
- Consumers need to know the strengths and limitations of benchmarks to properly use benchmark data
- New benchmarks will always be required
  - It is not necessary to "boil the ocean"
- Benchmark development organizations should share and learn from each other

Trademarks and Disclaimers

The opinions expressed in this paper are those of the author and do not represent official views of either the Transaction Processing Performance Council or the IBM Corporation