Types of Workloads

- Terminology
- Test Workloads for Computer Systems
  - Addition Instruction
  - Instruction Mixes
  - Kernels
  - Synthetic Programs
  - Application Benchmarks: Sieve, Ackermann's Function, Debit-Credit, SPEC

Part II: Measurement Techniques and Tools

Measurements are not to provide numbers but insight

- Ingrid Bucher

1. What are the different types of workloads?
2. Which workloads are commonly used by other analysts?
3. How are the appropriate workload types selected?
4. How is the measured workload data summarized?
5. How is the system performance monitored?
6. How can the desired workload be placed on the system in a controlled manner?
7. How are the results of the evaluation presented?

Terminology

- **Test workload**: Any workload used in performance studies. Test workload can be real or synthetic.
- **Real workload**: Observed on a system being used for normal operations.
- **Synthetic workload**:
  - Similar to real workload
  - Can be applied repeatedly in a controlled manner
  - No large real-world data files
  - No sensitive data
  - Easily modified without affecting operation
  - Easily ported to different systems due to its small size
  - May have built-in measurement capabilities.

Test Workloads for Computer Systems

1. Addition Instruction
2. Instruction Mixes
3. Kernels
4. Synthetic Programs
5. Application Benchmarks

Addition Instruction

- Processors were the most expensive and most used components of the system
- Addition was the most frequent instruction

Instruction Mixes

- Instruction mix = instructions + usage frequency
- Gibson mix: Developed by Jack C. Gibson in 1959 for IBM 704 systems.

1.	Load and Store	31.2
2.	Fixed-Point Add and Subtract	6.1
3.	Compares	3.8
4.	Branches	16.6
5.	Floating Add and Subtract	6.9
6.	Floating Multiply	3.8
7.	Floating Divide	1.5
8.	Fixed-point Multiply	0.6
9.	Fixed-point Divide	0.2
10.	Shifting	4.4
11.	Logical, And, Or, etc.	1.6
12.	Instructions Not Using Registers	5.3
13.	Indexing	18.0
	Total	100.0

Instruction Mixes (Cont)

- Disadvantages:
  - Complex classes of instructions not reflected in the mixes.
  - Instruction time varies with:
    - Addressing modes
    - Cache hit rates
    - Pipeline efficiency
    - Interference from other devices during processor-memory access cycles
    - Parameter values
    - Frequency of zeros as a parameter
    - The distribution of zero digits in a multiplier
    - The average number of positions of preshift in floating-point add
    - Number of times a conditional branch is taken

Instruction Mixes (Cont)

- Performance Metrics:
  - MIPS = Millions of Instructions Per Second
  - MFLOPS = Millions of Floating Point Operations Per Second

Kernels

- Kernel = nucleus
- Kernel= the most frequent function
- Commonly used kernels: Sieve, Puzzle, Tree Searching, Ackerman's Function, Matrix Inversion, and Sorting.
- Disadvantages: Do not make use of I/O devices

Synthetic Programs

- To measure I/O performance lead analysts $\Rightarrow$ Exerciser loops
- The first exerciser loop was by Buchholz (1969) who called it a synthetic program.
- A Sample Exerciser: See program listing Figure 4.1 in the book

Synthetic Programs

- Advantage:
  - Quickly developed and given to different vendors.
  - No real data files
  - Easily modified and ported to different systems.
  - Have built-in measurement capabilities
  - Measurement process is automated
  - Repeated easily on successive versions of the operating systems
- Disadvantages:
  - Too small
  - Do not make representative memory or disk references
  - Mechanisms for page faults and disk cache may not be adequately exercised.
  - CPU-I/O overlap may not be representative.
  - Loops may create synchronizations $\Rightarrow$ better or worse performance.

Application Benchmarks

- For a particular industry: Debit-Credit for Banks
- Benchmark = workload (Except instruction mixes)
- Some Authors: Benchmark = set of programs taken from real workloads
- Popular Benchmarks

Sieve

- Based on Eratosthenes' sieve algorithm: find all prime numbers below a given number *n*.
- Algorithm:
  - Write down all integers from 1 to n
  - Strike out all multiples of k, for k=2, 3, …, $\sqrt{n}$.
- Example:
  - Write down all numbers from 1 to 20. Mark all as prime:

  <u>1</u>, <u>2</u>, <u>3</u>, <u>4</u>, <u>5</u>, <u>6</u>, <u>7</u>, <u>8</u>, <u>9</u>, <u>10</u>, <u>11</u>, <u>12</u>, <u>13</u>, <u>14</u>, <u>15</u>, <u>16</u>, <u>17</u>, <u>18</u>, <u>19</u>, <u>20</u>
- Remove all multiples of 2 from the list of primes:

<u>1</u>, <u>2</u>, <u>3</u>, 4, <u>5</u>, 6, <u>7</u>, 8, <u>9</u>, 10, <u>11</u>, 12, <u>13</u>, 14, <u>15</u>, 16, <u>17</u>, 18, <u>19</u>, 20

Sieve (Cont)

- The next integer in the sequence is 3. Remove all multiples of 3:

<u>1</u>, <u>2</u>, <u>3</u>, 4, <u>5</u>, 6, <u>7</u>, 8, 9, 10, <u>11</u>, 12, <u>13</u>, 14, 15, 16, <u>17</u>, 18, <u>19</u>, 20

- $5 \geq \sqrt{20} \Rightarrow$ Stop
- Pascal Program to Implement the Sieve Kernel: See Program listing Figure 4.2 in the book

Ackermann's Function

- To assess the efficiency of the procedure-calling mechanisms. The function has two parameters and is defined recursively.
- Ackermann(3, *n*) evaluated for values of *n* from one to six.
- Metrics:
  - Average execution time per call
  - Number of instructions executed per call, and
  - Stack space per call
- Verification: Ackermann(3, *n*) = $2^{n+3}-3$
- Number of recursive calls in evaluating Ackermann(3,*n*):

$$(512\times 4^{n-1} - 15 \times 2^{n+3} + 9n + 37)/3$$

Execution time per call.

- Depth of the procedure calls = $2^{n+3}-4 \Rightarrow$ stack space required doubles when $n \leftarrow n+1$.

Ackermann Program in Simula

- See program listing Figure 4.3 in the book

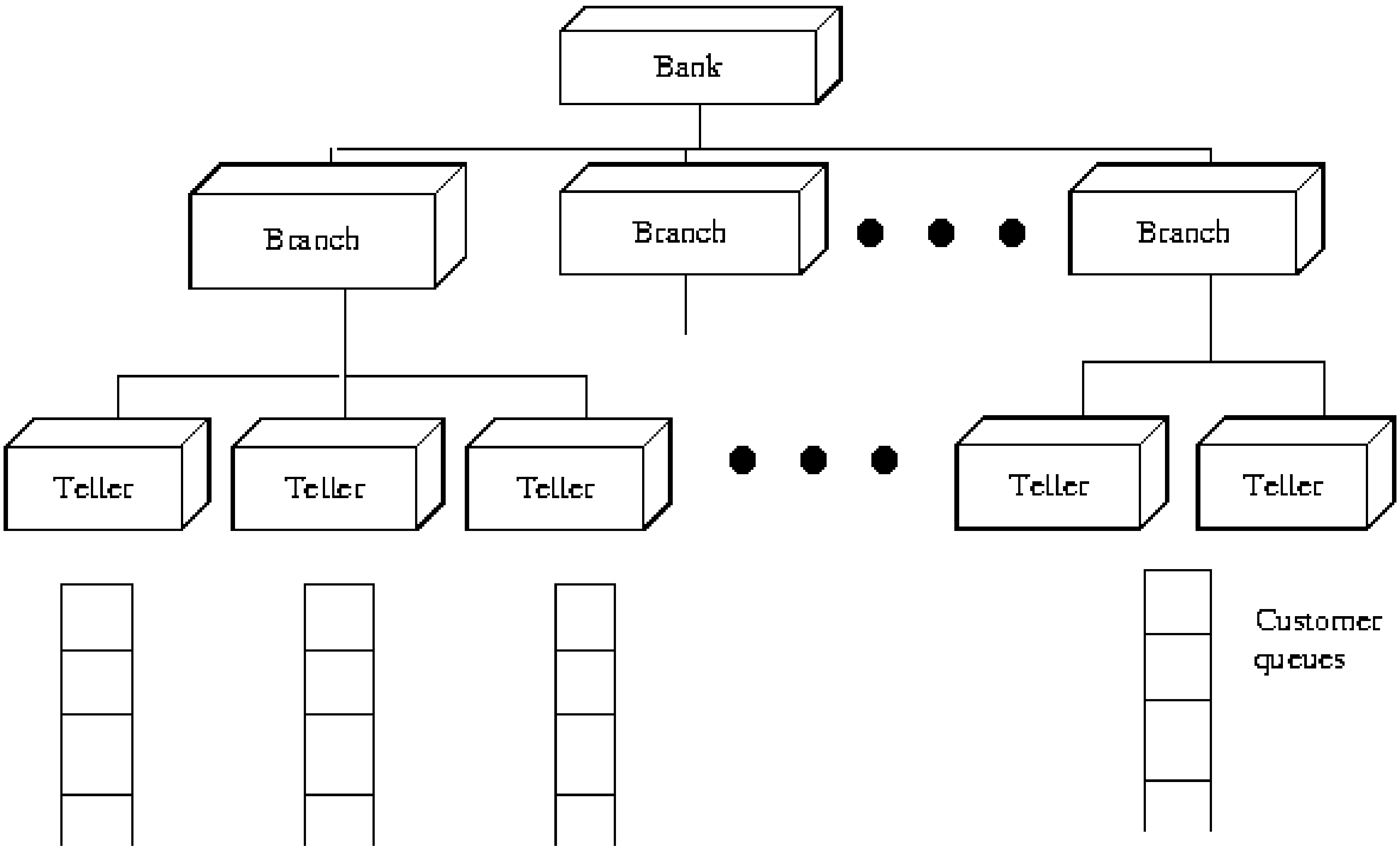
Other Benchmarks

- Whetstone
- U.S. Steel
- LINPACK
- Dhrystone
- Doduc
- TOP
- Lawrence Livermore Loops
- Digital Review Labs
- Abingdon Cross Image-Processing Benchmark

Debit-Credit Benchmark

- A de facto standard for transaction processing systems.
- First recorded in Anonymous et al (1975).
- In 1973, a retail bank wanted to put its 1000 branches, 10,000 tellers, and 10,000,000 accounts online with a peak load of 100 Transactions Per Second (TPS).
- Each TPS requires 10 branches, 100 tellers, and 100,000 accounts.

Debit-Credit (Cont)



Debit-Credit Benchmark (Continued)

- Metric: price/performance ratio.
- Performance: Throughput in terms of TPS such that 95% of all transactions provide one second or less response time.
- Response time: Measured as the time interval between the arrival of the last bit from the communications line and the sending of the first bit to the communications line.
- Cost = Total expenses for a five-year period on purchase, installation, and maintenance of the hardware and software in the machine room.
- Cost does not include expenditures for terminals, communications, application development, or operations.

Pseudo-code Definition of Debit-Credit

- See Figure 4.5 in the book
- Four record types: account, teller, branch, and history.
- Fifteen percent of the transactions require remote access
- Transactions Processing Performance Council (TPC) was formed in August 1988.
- TPC Benchmark™ A is a variant of the debit-credit
- Metrics: TPS such that 90% of all transactions provide two seconds or less response time.

SPEC Benchmark Suite

- Systems Performance Evaluation Cooperative (SPEC): Non-profit corporation formed by leading computer vendors to develop a standardized set of benchmarks.
- Release 1.0 consists of the following 10 benchmarks: GCC, Espresso, Spice 2g6, Doduc, LI, Eqntott, Matrix300, Fpppp, Tomcatv
- Primarily stress the CPU, Floating Point Unit (FPU), and to some extent the memory subsystem $\Rightarrow$ To compare CPU speeds.
- Benchmarks to compare I/O and other subsystems may be included in future releases.

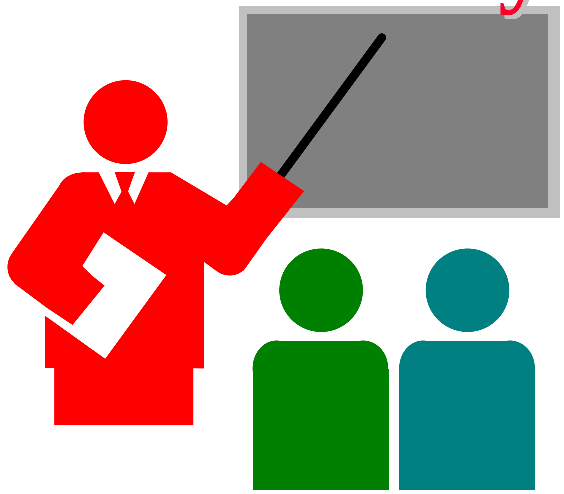
SPEC (Cont)

- The elapsed time to run two copies of a benchmark on each of the N processors of a system (a total of 2N copies) is measured and compared with the time to run two copies of the benchmark on a reference system (which is VAX-11/780 for Release 1.0).
- For each benchmark, the ratio of the time on the system under test and the reference system is reported as **SPECthruput** using a notation of #CPU@Ratio. For example, a system with three CPUs taking 1/15 times as long as the the reference system on GCC benchmark has a SPECthruput of 3@15.
- Measure of the per processor throughput relative to the reference system

SPEC (Cont)

- The aggregate throughput for all processors of a multiprocessor system can be obtained by multiplying the ratio by the number of processors. For example, the aggregate throughput for the above system is 45.
- The geometric mean of the SPECthruputs for the 10 benchmarks is used to indicate the overall performance for the suite and is called **SPECmark**.

Summary



- Synthetic workload are representative, repeatable, and avoid sensitive information
- Add instruction – most frequent instruction initially
- Instruction mixes, Kernels, synthetic programs
- Application benchmarks: Sieve, Ackerman, …
- Benchmark standards: Debit-Credit, SPEC

Exercise 4.1

Select an area of computer systems (for example, processor design, networks, operating systems, or databases), review articles on performance evaluation in that area and make a list of benchmarks used in those articles.

Exercise 4.2

Implement the Sieve workload in a language of your choice, run it on systems available to you, and report the results.