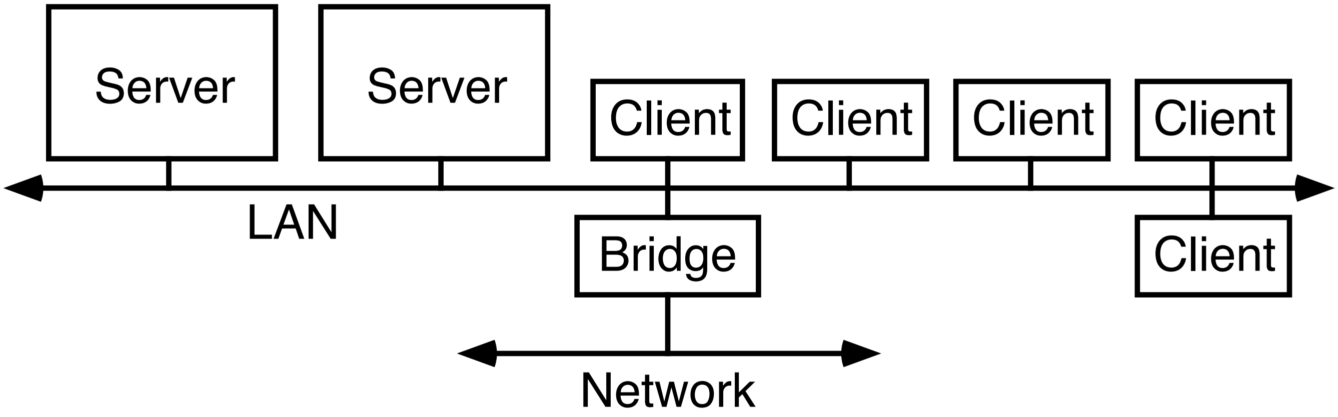
Topics in computer architecture

Benchmarking

P.J. Drongowski
SandSoftwareSound.net

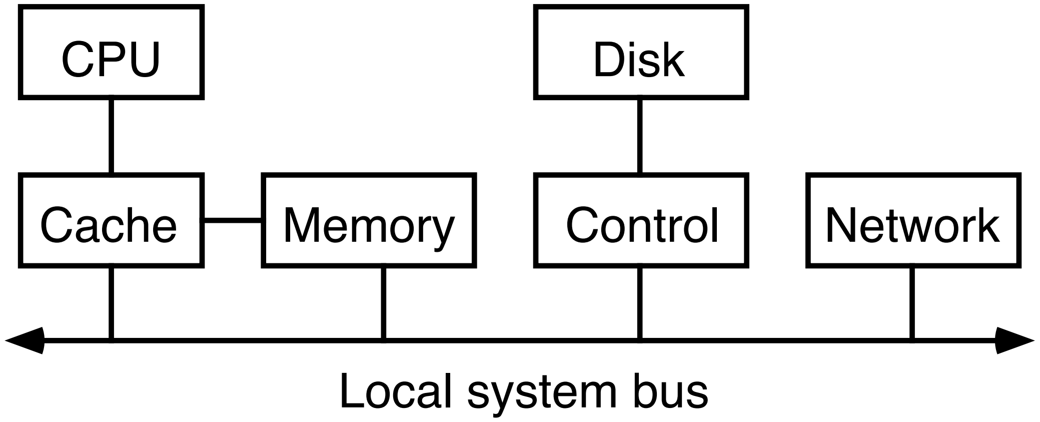
Network level



- Resource characterization
  - Server
    - Processing throughput
    - Throughput of secondary storage
    - OS issues: buffers, scheduling, etc.
  - Client
    - Processing throughput
    - OS issues: memory management, etc.
  - Network
    - Bandwidth (burst, sustained)
    - Error detection and recovery
    - Transfer protocol

- Workload characterization
  - Electronic office
  - CAD/CAE
  - Database transactions
  - Industrial process control

Workstation level



- Resource characterization
  - CPU
    - Throughput
    - Basic machine/memory cycle speed
  - Cache memory
    - Caching scheme (direct map, etc.)
    - Size
    - Memory speed vs. density
  - Memory
    - Size (standard SIMM's, etc.)
    - Memory speed vs. density
  - Secondary storage
    - Control complexity
    - Data caching
    - Error detection and correction
  - Local system bus
    - Bandwidth (peak, sustained)
    - Arbitration (synchronous vs. asynchronous)
    - Standard versus proprietary

- Workload characterization
  - Graphics display / human interface
  - Application task

Benchmarks

- Millions of instructions per second (MIPS-rating)
  - "Meaningless indicator of performance"
  - "VUP" ⇔ "VAX unit of performance"

- Artificial benchmark programs
  - Integer performance (Dhrystone)
  - Floating performance (Whetstone)
  - Input / output performance
  - Compiler / OS variability

- Real application program is *most* conclusive

- Benchmarks
  - Dhrystone (integer)
  - Stanford STAN INT (integer)
    - Permute 7 elements 5 times
    - Towers of Hanoi
    - Eight queens problem 50 times
    - Multiply two 40 x 40 integer matrices
    - Soma Cube-type problem
    - Quick sort of 5000 elements
    - Bubble sort of 500 elements
    - Binary tree sort of 5000 items
  - MIPS Computer Systems Unix
    - Unix utilities
    - grep, yacc, diff and nroff
    - Tend to overstate performance
  - Whetstone (floating and integer)
  - Livermore Fortran kernels
  - LINPACK (floating, linear equations)
  - Spice (floating, electronic simulation)
  - Dudoc (floating, nuclear reactor simulation)

Floating point benchmarks

- Livermore Fortran Kernels
    - McMahon, "The Livermore Fortran Kernels: A computer test of the numerical performance range," Lawrence Livermore National Labs, December 1986.
    - Vectorizing supercomputer benchmarks
    - 24 "kernels" or pieces of code
        - Multiple runs on different data sets
        - Check for correct results
        - Verify timing accuracy
        - Report execution rates
    - Results
        - Not "distilled" into a single number
        - Rates for three different vector sizes

- LINPACK
    - Linear equations (multiplies and additions)
    - Time required to solve 100 x 100 system
    - Results reported in MFLOPS
    - Data set often exceeds cache memory size
    - Use of submatrix sensitive to cache size
        - Submatrix tends to spread data out
        - Depends on how data wraps around / overlaps
        - Example: 64Kbyte data cache
    - Benchmark is easily vectorized

Synthetic benchmarks

- Synthetic benchmark
  - Single program
  - Reflect frequency of source language constructs
  - Based on frequencies in actual programs
  - In compiled form, will reflect frequency of instructions
  - Will measure quality of compiler

- Whetstone benchmark
  - H.J. Curnow and B.A. Wichman, "A synthetic benchmark," Computing Journal, Vol. 19, No. 1, February 1976, pg. 43-49
  - Originally written in Algol
  - Reflects numerical computing, floating-point arithmetic
  - Statement type distribution was collected in 1970
  - Constructs may be outdated
  - Does not consider locality of reference

- Dhrystone benchmark
  - Reinhold P. Weicker, "Dhrystone: A synthetic systems programming benchmark, CACM, Vol. 27, No. 10, October 1984, pg. 1013 - 1030
  - System programming oriented
  - Enumeration, record and pointer datatypes
  - Originally coded in Ada™ (Pascal and C)
  - Fequencies based on recent measurements
  - See paper for extensive distribution tables

Distributions

- Static statistics
  - Assignment
  - Call (user / standard procedures)
  - Return
  - If (with else / without else)
  - Loop with condition (while / repeat)
  - Loop with "for"
  - With
  - Case
  - Exit loop
  - Goto
  - Other

- Dynamic statistics (harder to collect)
  - Breakdown of assignment statements
    - Left-hand side
      - Variable
      - Array element
      - Record component
      - Other
    - Right-hand side
      - Constant
      - Variable
      - Array element
      - Function result
      - Record component
      - Other
  - Operator use
    - Arithmetic: `+, -, *, /, div, modulus`
    - Comparison: `==, !=,` etc.
    - Logic: `AND, OR, NOT`
  - Number of parameters
  - Operand types (final, first, all on access path)
  - Operand locality (global, local, parameter, constant)

Dhrystone comments

- Does not use:
    - Floating point
    - I/O or operating system calls
    - Code that can be vectorized
- Reported in Dhrystones per second

VAX 11/780	1,600
Sun 3/80	5,150
VAX 8600	7,400
IBM 3081	15,000
CRAY X-MP	18,500
Sun SPARCstation 1	22,050
DECstation 3100	25,000
Apollo DN10000	25,550
Sun SPARCsystem 300	27,750
IBM 3090/200	31,250

- Unusual benchmark features
    - Low number of instructions per function call
        - 35-40 instructions on MIPS machines
        - 50-60 or higher on real C programs
    - Favors lean function call (e.g., against VAX)
    - Nesting depth is low (3-4 levels)
    - 30-40% in `strcpy` on atypically long strings
- Very sensitive to compiler optimizations
    - Interprocedural register allocation
    - Procedure inlining
    - Inline expansion of strcpy
- Small size; Dhrystone fits into most instruction caches

TABLE VIII. Assignment Statements in "Dhrystone"

		Percentage
$V1 := V2$ (incl. $V1 := F$ (. .))	10	
$V :=$ Constant	12	
Assignment, with array element	7	
Assignment, with record component	6	
	$\overline{35}$	35
$X := Y + \vert - \vert$ and $\vert$ or Z	5	
$X := Y + \vert - \vert$ "=" Constant	6	
$X := X + \vert - 1$	3	
$X := Y * \vert / Z$	2	
$X :=$ Expression, two operators	1	
$X :=$ Expression, three operators	1	
	$\overline{18}$	18
		$\overline{53}$

TABLE IX. Control Statements in "Dhrystone"

			Percentage	
if · · · · then · · · ·			14	
with "else"	7			
without "else"	7			
executed		3		
not executed		4		
for I in 1 · · N loop · · ·			6	counted every time the loop condition is evaluated
while · · · loop · · ·			4	
loop · · · exit when $A = B$ · ·			1	
case · · · is			1	
return			5	
rename			$\underline{1}$	
			32	

TABLE X. Call Statements in "Dhrystone"

	Percentage	
$P(\cdots)$		
procedure call		10
same package	5	
other package	5	
$X := F(\cdots)$		
function call		5
same package	2	
other package	3	
		15

TABLE XI. Distribution of Operators in "Dhrystone"

	Number	Percentage
Arithmetic	27	52.9
+	16	31.4
−	7	13.7
*	3	3
/ (int div)	2.0	1
Comparison	39.2	20
=	9	17.6
/ =	4	7.8
>	1	2.0
<	3	5.9
> =	1	2.0
< =	2	3.9
Logic	4	7.8
AND	1	2.0
OR	1	2.0
NOT	2	3.9
Sum	99.9	51

TABLE XII. Distribution of Operand Types in "Dhrystone"

	Number	Percentage
Integer	131	54.4
Character	47	19.5
Enumeration	30	12.4
Boolean	11	4.6
Pointer (Access Type)	12	5.0
String_30	6	2.5
Array	2	0.8
Record	2	0.8
	241	100.0

TABLE XIII. Distribution of Locality of Operands in "Dhrystone"

	Number	Percentage
Local variables	117	48.5
Global variables	19	7.9
same package	18	7.5
other package	1	0.4
Parameters	45	18.7
in	27	11.2
in out	12	5.0
out	6	2.5
Function results	5	2.1
Constants	55	22.8
	241	100.0