

Analisa Kinerja Sistem

Tipe Workload

1

- ### Tipe Workload
- *Test workload* - workload yang digunakan untuk studi kinerja
 - *Real workload* - diobservasi pada sistem yang sedang digunakan.
 - Tidak dapat diulang (dengan mudah)
 - Mungkin tidak pernah ada (rencana sistem)
 - *Synthetic workload* - karakteristik serupa untuk real workload
 - Dapat diterapkan berulang kali
 - Relatif mudah untuk di terapkan
 - *Benchmark == Workload*
 - Benchmarking adalah proses membandingkan 2+ sistem dengan workload tertentu
- 2

- ### Outline
- Introduksi (sudah)
 - Instruksi Penjumlahan (berikut)
 - Mix instruksi
 - Kernel
 - Program sintetik
 - Benchmarks Aplikasi
- 3

- ### Instruksi Penjumlahan
- Komputer awalnya memiliki CPU sebagai komponen termahal
 - Operasi dengan frekuensi terbanyak adalah penjumlahan
 - Komputer dengan instruksi penjumlahan yang lebih cepat berkinerja lebih baik
 - Maka, jalankan sejumlah operasi penjumlahan sebagai test workload
 - Problem
 - Lebih banyak instruksi digunakan
 - Beberapa lebih rumit dari yang lainnya
- 4

- ### Mix Instruksi
- Jumlah dan kompleksitas instruksi meningkat
 - Dapat mengukur instruksi secara individuak, tetapi menggunakan jumlah yang berbeda
 - Mengukur frekuensi relatif berbagai instruksi pada sistem real
 - Menggunakan sebagai faktor pembebanan untuk mendapatkan waktu instruksi rata-rata
 - *Campuran Instruksi*
 - * Unit adalah
 - Millions of Instructions Per Second (MIPS)
 - Millions of Floating-Point Ops per Sec (MFLOPS)
- 5

Contoh: Mix Instruksi Gibson

1. Load and Store	13.2
2. Fixed-Point Add/Sub	6.1
3. Compares	3.8
4. Branches	16.6
5. Float Add/Sub	6.9
6. Float Multiply	3.8
7. Float Divide	1.5
8. Fixed-Point Multiply	0.6
9. Fixed-Point Divide	0.2
10. Shifting	4.4
11. Logical And/Or	1.6
12. Instructions not using regs	5.3
13. Indexing	<u>18.0</u>
Total	100

6

1959, IBM 650 IBM 704

Problem dengan Mix Instruksi

- Dalam sistem modern, variabel waktu instruksi tergantung pada
 - Modus pengalamatan, cache hit rates, pipelining
 - Interferensi dengan divais lain selama akses prosesor-memori
 - Distribusi nol dalam multiplier
 - Waktu yang dibutuhkan untuk percabangan kondisional
- Mix tidak merefleksikan hardware khusus seperti page table lookups
- Hanya merepresentasikan kecepatan prosesor
 - Bottleneck mungkin dilain bagian dari sistem

7

Kernels

- Menggunakan set instruksi yang membentuk service yang disediakan oleh prosesor. *Kernel*.
 - Awalnya, tidak mempertimbangkan I/O, dan juga sering disebut *processing kernel*
- Set operasi untuk problem
 - Ex: Sieve, Tree Searching, Inversi Matriks
- Beberapa problem seperti zero dan percabangan tidak diterapkan
- Problem
 - I/O tetap tidak dipertimbangkan

8

Program Sintetik

- Menambahkan request I/O ke load uji
 - Menambahkan loop kontrol sehingga dapat menentukan frekuensi request sesuai kebutuhan
 - Mudah diturunkan, diterapkan
 - Dapat memiliki data pengukuran langsung
- Akan tetapi
- Masih, tidak memperhatikan penggunaan memori representatif atau akses disk
 - Terkadang kecil, sehingga, tidak menyentuh memori virtual

9

Contoh Program Pembangkit Workload Sintetik

Buckholz, 1969

```
DIMENSION Record(500)
!Control parameters
Num_Computes=500      !Repeat count for computation
Num_Reads=50         !Number of records read
Num_Writes=40        !Number of records written
Num_Iterations=1000  !Repeat count for the experiment

!Open files
OPEN(UNIT=1, NAME='In.dat', TYPE='DISK',
      IFORM='FORMATTED', ACCESS='SEQUENTIAL')
OPEN(UNIT=2, NAME='Out.dat', TYPE='REW',
      IFORM='UNFORMATTED', ACCESS='SEQUENTIAL')
WRITE(UNIT=1, RECORD=1, I=1, J=1, K=1)
WRITE(UNIT=2, RECORD=1, I=1, J=1, K=1)
DO 100 I=1, Num_Iterations
  *****
  !Perform a number of read I/O's
  DO 200 J=1, Num_Reads
    READ(UNIT=1, RECORD=J)
  *****
  !Do computation
  DO 300 K=1, Num_Writes
    *****
    !Perform a number of write I/O's
    WRITE(UNIT=2, RECORD=K)
  *****
300 CONTINUE
100 CONTINUE
!Close files
CLOSE(UNIT=1)
CLOSE(UNIT=2)
!CPU time=(CPU2-CPU1)/Num_Iterations
Elapsed_Time=(Elapsed2-Elapsed1)/Num_Iterations
TYPE *, 'CPU time per iteration is ', CPU_Time
TYPE *, 'Elapsed time per iteration is ', Elapsed_Time
*****
END
```

10

Workload Aplikasi

- Untuk sistem kegunaan-khusus, mungkin dapat menjalankan aplikasi representatif sebagai pengukuran kinerja
 - Mis: reservasi penerbangan
 - Mis: banking
- Meenggunakan keseluruhan sistem (I/O, dll).
- Isu meliputi
 - Parameter input
 - multiuser
- Hanya dapat diterapkan ketika aplikasi tertentu yang ditargetkan

11

Benchmark Populer: Sieve (1/2)

- Sieve of Eratosthenes (menemukan bilangan prima)
- Tuliskan semua bilangan 1 n
- Keluarkan kelipatan k untuk $k = 2, 3, 5 \dots$ akar(n)
 - Sisanya adalah bilangan prima

12

Benchmark Populer: Sieve (2/2)

1. Write down all numbers from 1 to 20.

Mark all as prime:

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

2. Remove all multiples of 2 from the list of primes:

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

3. The next integer in the sequence is 3.

Remove all multiples of 3:

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

4. $5 \geq \sqrt{20} \Rightarrow$ Stop

13

Benchmark Populer: Fungsi Ackermann (1/2)

- Menilai efisiensi mekanisme pemanggilan prosedur
- Fungsi Ackermann memiliki dua parameter, yaitu rekursif
 - Benchmark memanggil Ackermann(3, n) untuk nilai n = 1 sampai 6
- Return value adalah $2^{m-3} - 3$, dapat digunakan untuk memverifikasi implementasi
- Banyaknya call:

$$(512 \times 4^{n-1} - 15 \times 2^{n-3} + 9n + 37) / 3$$
 - Dapat digunakan untuk menghitung waktu per call
- Kedalaman adalah $2^{m-3} - 4$, ruang stack ganda $n++$

14

Benchmark Populer: Fungsi Ackermann (2/2)

(Simula)

```

BEGIN
  INTEGER n;          'Loop index;
  INTEGER j;          'Function value;
  INTEGER num_calls;  'Number of recursive calls;
  INTEGER k;          'Contains 2**(n-3);
  INTEGER k1;         'Contains 4**(n-1);
  REAL t1,t2;         'CPU time values;

  INTEGER PROCEDURE Ackermann(n,n); VALUE m,n; INTEGER m,n;
  Ackermann := IF n=0 THEN n+1
                ELSE IF n=0 THEN Ackermann(n-1,1)
                ELSE Ackermann(n-1,Ackermann(m,n-1));
'Main Program:
  k := 16; k1 := 1;  'Initialize k and k1 for n=1;
  FOR n := 1 STEP 1 UNTIL 6 DO
  BEGIN
    t1 := CPU TIME;      'Beginning CPU time;
    j := Ackermann(3,n); 'Compute the function;
    t2 := CPU TIME;      'Ending CPU time;
    IF j <= k-3 THEN OUTPUT('Wrong Value');
    OUTPUT('Set CPU Time for Ackermann(3,');
    OUTPUT(n,1); OUTPUT(') is');
    OUTPUTREAL(t2-t1,7,15); OUTPUT;
    num_calls := (512*j-15*n+9*n+37)/3;
    OUTPUT('CPU Time per call:');
    OUTPUTREAL((t2-t1)/num_calls,7,15);
  END;
  k1 := 4*k1;          'Update k1 for the next n;
  k := 2*k;            'Update k for the next n;
END
END
    
```

15

Benchmark Populer: Whetstone

- Set 11 modul didesain untuk menentukan frekuensi terobservasi dalam program ALGOL
 - Array addressing, arithmetic, subroutine calls, parameter passing
 - Diturunkan ke Fortran, paling populer dalam C, ...
- Banyak variasi Whetstone, maka hati-hati dalam membandingkan hasil
- Problem - kernel spesifik
 - Hanya berlaku untuk aplikasi kecil, scientific (floating) yang muat dalam cache
 - Tidak menguji I/O

16

Benchmark Populer: LINPACK

- Program yang menyelesaikan sistem penuh dengan persamaan linier
 - Banyak penjumlahan dan perkalian float
 - Intinya adalah Basic Linear Algebra Subprograms (BLAS), dipanggil berulang-ulang
- Biasanya, menyelesaikan sistem persamaan 100x100
- Merepresentasikan aplikasi mechanical engineering pada workstation
 - Gambaran kasar untuk analisa elemen terbatas
 - Kecepatan komputasi tinggi dan pemrosesan grafik yang baik

17

Benchmark Populer: Dhrystone

- Targetnya untuk merepresentasikan lingkungan pemrograman sistem
- Paling umum dalam C, tetapi dgn banyak versi
- Kedalaman bersarang dan instruksi yang rendah dalam tiap call
- Banyak waktu untuk meng-kopi string
- Kebanyakan kinerja integer tanpa operasi float

18

Benchmark Populer: Lawrence Livermore Loops

- 24 uji scientific
- Operasi *floating point*
 - Aplikasi fisika dan kimia ditemukan operasi floating point 40-60%
- Relevan untuk: dinamika fluida, desain pesawat terbang, pemodelan cuaca

19

Benchmark Populer: Debit-Credit

- Standard Defacto untuk Sistem Pemrosesan Transaksi
- Bank ritel menginginkan 1000 cabang, 10k teller, 10000K account online dengan load puncak 100 TPS
- Kinerja dalam TPS dimana 95% transaksi memiliki waktu respons 1 detik atau kurang (kedatangan bit terakhir, pengiriman bit pertama)
- Sekarang Transaction Processing Council (TPC) memiliki benchmark yang lebih presisi
 - TPC-A, TPC-B, TCP-C

20

Benchmark Populer: SPEC

- Systems Performance Evaluation Cooperative (SPEC) (<http://www.spec.org>)
 - Vendor komputer terkemuka, non-profit
 - Kumpulan benchmarks
- CPU2000: CPUINT dan CPUFP
 - Pembuatan CPU2004
- Grafik
- Sistem dan Aplikasi:
 - Web, Java Client-Server, Network Files System, Mail
- Database hasil
- Membandingkan kinerja dengan mesin baseline

21