

Bab 2 Jenis-jenis Metriks

Metriks atau yang disebut dengan variabel kinerja merupakan variabel atau besaran yang dipilih dan diukur sebagai representasi dari kinerja sistem. Tapi tidak semua kinerja sistem perlu diukur, kinerja sistem yang bersifat kewajaran tidak perlu diukur. Sebagai contoh komputer yang memiliki single cache memory dengan virtual harddisk yang kecil maka saat beberapa aplikasi dibuka sekaligus maka kecepatannya menjadi lambat, prosesoranya menjadi sibuk dan komputer terdiam. Ini suatu kewajaran yang tidak perlu diukur, kinerja yang seperti ini disebut kinerja bawaan (*inherited performance*). Komputer menjadi panas saat dipakai, inipun bukan suatu kinerja yang perlu diukur.

Ciri suatu variabel dapat dipilih sebagai metriks yaitu variabel yang merepresentasikan kinerja adalah :

- a. Saat beban sistem berubah maka metriks yang dipilih juga berubah
- b. Saat beban naik atau turun ada suatu pola kinerja yang bisa dimonitoring
- c. Secara terpisah (*independence*) atau bersama-sama perubahan metriks merepresentasikan besaran kualitas.

Memilih komputer berdasarkan kinerja merupakan hal yang penting saat memilih komputer yaitu dengan membanding metrik dari dua atau lebih komputer. Sudut pandang dari pengguna adalah komputer mana yang kinerjanya terbaik, harganya termurah dan perbandingan antara harga dan kinerjanya sesuai. Sedangkan dari perspektif perancang komputer yang dihadapkan dengan permasalahan perancangan berupa kinerja terbaik mana yang berhadil ditingkatkan, harga termurah dan juga perbandingan harga dan kinerja yang terbaik. Baik pengguna maupun perancangan melakukan hal yang sama yaitu perbandingan acuan (*benchmark*) dan evaluasi metriks (pengukuran kinerja).

Setiap arsitektur komputer yang berbeda biasa dianalisis berdasarkan dari biayanya dan kinerja yang dihasilkan. Terkadang kita perlu memahami :

- a. apakah perlu mengganti perangkat keras atau sebenarnya hanya mengganti sistem operasinya untuk meningkatkan kinerja
- b. serta mengapa pada arsitektur yang berbeda kinerjanya juga berbeda saat menjalankan program yang sama;
- c. Bagaimana intruksi mesin mempengaruhi kinerja.

2.1. Metriks sebagai penentu Kinerja Komputer

Metriks yang sering digunakan untuk mengevaluasi kinerja komputer adalah waktu respon (*Response Time*) dan *Throughput*. Waktu respon atau waktu eksekusi adalah waktu dari awal suatu program mulai sampai selesai. Pada multi tasking mungkin saja suatu program dipanggil-panggil menjadi beberapa task dan eksekusi masing-masing task dipengaruhi oleh penjadwalannya. Sedangkan *throughput* adalah jumlah pekerjaan yang selesai dalam kurun waktu tertentu. Biasanya manajer data lebih peduli terhadap *throughput* daripada waktu respon. seorang manajer data ingin agar *throughput*nya meningkat atau dengan kata lain jumlah pekerjaan yang selesai lebih banyak untuk suatu kurun waktu yang sama. Sedangkan waktu respon diharapkan lebih berkurang atau semakin cepat selesai tasknya. Sebagai ilustrasi pada pabrik mobil, diperlukan 4 jam untuk membuat mobil, ini adalah waktu respon; dan 6 mobil per jam yang dihasilkan , ini adalah *throughput*.

2.1.1. Waktu Eksekusi

Terdapat beberapa metriks yang terkait dengan waktu eksekusi dalam Teknik Komputer yaitu *elapsed Time* (*Response Time*) dan *CPU Time*. *Elapsed Time* digunakan untuk menghitung waktu respon hampir di setiap perangkat komputer seperti untuk menghitung pada Disk, akses memori, alat I/O. Metriks yang berguna namun tidak cocok untuk dijadikan perbandingan saat terjadi multitasking. Sedangkan *CPU Time* tidak menghitung waktu yang diselesaikan oleh perangkat I/

O, sesuai dengan namanya ini adalah waktu yang diperlukan oleh komputer untuk melaksanakan setiap instruksi. CPU Time dapat dibagi menjadi waktu sistem (System Time) untuk sistem operasi dan Waktu Pengguna (User Time) untuk program.

Untuk satu program yang dijalankan di komputer X, kinerja komputer X dapat dinyatakan sebagai :

$$Kinerja_x = 1 / Waktu Eksekusi_x$$

Komputer X lebih cepat n kali terhadap Y jika: Speedup = n

$$Speedup = Kinerja_x / Kinerja_y = Waktu Eksekusi_y / Waktu Eksekusi_x$$

Mesin komputer dikatakan memiliki kinerja yang lebih baik jika total waktu eksekusi untuk programnya lebih pendek dibandingkan komputer lain.

Sebagai contoh:

Sebuah program yang sama dikerjakan oleh dua mesin komputer yang berbeda, komputer A dan komputer B. Waktu eksekusi untuk komputer A adalah 1 detik, sedangkan waktu eksekusi untuk komputer B adalah 10 detik. Maka

$$\frac{Kinerja_A}{Kinerja_B} = \frac{Waktu Eksekusi_B}{Waktu Eksekusi_A} = \frac{10}{1} = 10$$

Jadi kinerja komputer A lebih cepat 10 kali dari komputer B

2.1.2. Analisis CPU Time

CPU Time tergantung pada program yang dieksekusi, karena program merupakan kumpulan instruksi, maka CPU Time tergantung pada jumlah dan jenis instruksi-instruksi yang berada di dalam program tersebut. Pewaktuan komputer menggunakan clock rate dengan satuan Hz. Clock Rate ini merupakan waktu dari periode siklus yang disebut dengan clock cycle time (detik).

$$clock \ cycle \ time = \frac{1}{clock \ rate}$$

Biasanya dalam teknik komputer penggunaan eksekusi time dalam detik jarang digunakan untuk perhitungan waktu CPU tetapi lebih seringnya menggunakan siklus (cycle).

$$CPU \ Time = \frac{detik}{program} = \frac{siklus}{program} \times \frac{detik}{siklus}$$

Clock time : Detik per siklus adalah waktu diantara Clock Tick

Misal untuk komputer dengan clock rate 4 GHz maka clock timenya adalah 1/4.10⁹ detik atau 250 picodetik= (1/4.10⁹)x 10¹²

2.1.3. Bagaimana meningkatkan kinerja komputer

Misal suatu program diselesaikan dalam 10 detik pada komputer A yang clock ratenya 4GHz. Agar dapat diselesaikan 6 detik pada komputer B, namun komputer B memiliki teknologi yang clock cyclenya 1,2 kali clock Cyle A. Makan clock rate B dapat ditentukan sebagai berikut:

$$CPU \ Time_A = \frac{CPU \ clock \ Cycle_A}{clock \ rate_A}$$

$$10 \ detik = \frac{CPU \ clock \ cycle_A}{4 \times 10^9 \frac{cycle}{detik}}$$

$$\therefore CPU \ clock \ cycle_A = 40 \times 10^9 \ siklus$$

$$CPU \ Time_B = \frac{1,2 \times CPU \ clock \ cycle_A}{clock \ rate_B}$$

$$\therefore Clock \ rate_B = 8GHz$$

2.1.4. Mengukur waktu menggunakan Clock Cycle

Untuk menghitung clock cycle dipergunakan rata-rata clock cycle per intruksi yang disebut CPI (Clock cycle per Instruction) dengan persamaan sebagai berikut:

$$Clock \ cycle \ untuk \ satu \ program = Instruction \ count \times CPI$$

Instruction count adalah jumlah instruksi di dalam program.

Sehingga waktu eksekusi CPU untuk suatu program = Instruction count x CPI x clock cycle time

$$CPU \ Time = \frac{Instruksi}{program} \times \frac{siklus}{instruksi} \times \frac{detik}{siklus}$$

$$CPU \ Time = \frac{detik}{program}$$

CPI bercerita mengenai arsitektur instruksi, implementasi dari arsitektur komputer dan program yang diukur. Clock cycle besarnya sudah terdapat pada spesifikasi komputer, Instruction count diukur menggunakan simulator (software) atau menggunakan hardware counter pada register

Contoh:

Sebuah program dijalankan pada suatu komputer dengan parameter berikut:

Jumlah instruksi: 10.000.000 instruksi

Rata-rata CPI: 2,5 siklus/instruksi

CPU clock rate: 200 MHz

a. Berapakah waktu eksekusi untuk program ini:

$$\begin{aligned} CPU \ time &= \text{jumlah instruksi} \times CPI \times \text{Clock cycle} \\ &= 10.000.000 \times 2,5 \times 1 / \text{clock rate} \\ &= 10.000.000 \times 2,5 \times 5 \times 10^{-9} \\ &= 0,125 \text{ detik} \end{aligned}$$

b. Menggunakan program yang sama pada compiler yang lain diperoleh

instruction count: 9.500.000,

CPI baru: 3

CPU clock rate lebih cepat : 300 MHz

Berapa speedup terhadap compiler baru ini:

$$\begin{aligned} Speedup &= (10.000.000 \times 2,5 \times 5 \times 10^{-9}) / (9.500.000 \times 3 \times 3,33 \times 10^{-9}) \\ &= 0,125 / 0,095 = 1,32 \end{aligned}$$

atau 32% lebih cepat dengan compiler baru.

2.1.5. Waktu yang diperlukan instruksi

Tergantung pada jenis instruksinya, waktu yang diperlukan untuk eksekusi sebuah intruksi bisa berbeda, bahkan pada mesin komputer yang berbeda bisa didapati waktu eksekusi yang berbeda. Sehingga tidak bisa diambil kesimpulan satu siklus memiliki jumlah instruksi yang sama.

- Instruksi perkalian membutuhkan waktu eksekusi lebih lama dari instruksi penjumlahan.
- Operasi floating point lebih lama dari operasi bilangan integer
- Akses pada memori lebih lama dari akses pada register
- Mengganti cycle time akan mengubah jumlah siklus yang diperlukan oleh instruksi. Sebagai contoh program lama yang di-compile oleh Turbo Pascal versi DOS yang diperuntukan untuk komputer dengan clock rate 800 MHz akan terlalu cepat jika dieksekusi pada komputer di era sekarang yang kecepatannya di atas 1,6 GHz

Contoh :

Terdapat dua implementasi ISA (*Instruction Set Architecture*) yang sama.

Mesin A memiliki clock cycle time 250 ps dan CPI 2,0

Mesin B memiliki clock cycle time 500 ps dan CPI 1,2

Mesin komputer mana yang paling cepat untuk program ini? Dan seberapa cepat?

Jawab:

CPU clock cycleA = Instruction x 2,0

CPU clock cycle B = Instruction x 1,2

CPU timeA = CPU clock cycleA x CPU timeA = Instruction x 2,0 x 250ps = Instruction x 500ps

CPU timeB = CPU clock cycleB x CPU timeB = Instruction x 1,2 x 500ps = Instruction x 600 ps

$$\frac{CPU\ time_B}{CPU\ time_A} = 1,2$$

2.1.6. Frekuensi Instruksi dan CPI

Teknik analisa sistem komputer yang lain adalah menggunakan Frekuensi atau fraksi instruksi, pada teknik ini CPI merupakan nilai rata-rata untuk instruksi tertentu. Misal jika di dalam program ada dua jenis instruksi yaitu perkalian dan penjumlahan maka akan ada dua CPI yaitu CPI untuk instruksi perkalian dan CPI untuk instruksi penjumlahan.

Didefinisikan :

C_i = Jumlah Instruksi untuk jenis ke-i

CPI_i = CPI untuk instruksi jenis ke-i

F_i = Frekuensi atau fraksi instruksi jenis ke-i = $C_i / \text{Total instruksi} = C_i / I$

Maka:

$$CPI = \sum_{i=1}^n CPI_i \times F_i = CPI_1 \times F_1 + CPI_2 \times F_2 + \dots + CPI_n \times F_n$$