ISBN: 979-756-061-6

STUDI PENDAHULUAN PROSES PERHITUNGAN FRACTAL UNTUK PENGUKURAN UNJUK KERJA KOMPUTER

Ana Heryana dan Wawan Wardiana
Pusat Penelitian Informatika, Lembaga Ilmu Pengetahuan Indonesia
Komplek LIPI Gd 20 Lt 3 Jl. Sanguriang, Bandung, 40135

Abstrak

Unjuk kerja sistem komputer adalah kemampuan komputer dalam menyajikan informasi yang diperlukan oleh pemakai. Faktor-faktor yang menentukan unjuk kerja komputer adalah sekumpulan perangkat keras dan perangkat lunak. Unjuk kerja komputer dapat diukur dengan berbagai cara. Unjuk kerja komputer ditentukan oleh perangkat keras yang terpasang dan perangkat lunak yang dijalankan pada perangkat keras tersebut. Pengukuran dilakukan untuk mengetahui kemampuan komputer dalam mengolah dan menghasilkan informasi yang diperlukan oleh pemakai. Cara pengukuran yang paling banyak digunakan yaitu dengan menggunakan metoda benchmarking. Geometri fraktal mempunyai karakter-karakter penting antara lain self similar, self affine, self inverse, dan self squaring. Yang jelas skala panjangnya tidak spesifik atau invariant. Berbeda dengan geometri euklidean, penyekalaan fraktal dicirikan oleh bilangan-bilangan pecahan atau tak bulat (noninteger), yang disebut dimensi fraktal (fractal dimensions). Ciri-ciri yang biasanya dijumpai pada bangun fraktal, adalah bahwa bagian terkecil dari benda itu merupakan cerminan bentuk keseluruhannya (the part is reminiscent of the whole). Dengan kata lain, bahwa di dalam suatu himpunan fraktal, bagian dari himpunan tersebut merupakan skala kecil dari keseluruhannya.

***Kata kunci**: unjuk kerjakomputer, geometri fractal*

1. Pendahuluan

Pada era informasi ini, sumber daya komputer menjadi bagian yang tidak terpisahkan dari berbagai pekerjaan. Komputer digunakan sebagai alat bantu atau alat utama dalam pekerjaan menjadikannya sebagai komponen yang sangat penting, sehingga perlu dipertahankan kemampuan dan kestabilan unjuk kerjanya. Unjuk kerja komputer biasanya berbeda, sesuai dengan spesifikasi dari peralatan komputer tersebut, dan tentu saja disesuaikan dengan jenis pekerjaan yang ditangani oleh peralatan komputer tersebut.

Pengukuran unjuk kerja sistem komputer umumnya dilakukan dengan menggunakan dua cara berikut ini yaitu:

- Laju detak (*clock*) processor atau jumlah instruksi yang dapat diproses per satuan waktu yang diekspresikan dengan satuan MIPS (*Million Instruction Per Second*). Walaupun cara ini sangat memudahkan bagi pengguna komputer awam, namun menimbulkan ketidakjelasan dan ketidakadilan saat digunakan untuk membandingkan komputer yang berarsitektur RISC dan CISC. Dimana pada arsitektur RISC diperlukan banyak instruksi untuk melaksanakan suatu tugas dibandingkan CISC. Komputer RISC akan kelihatan lebih banyak dapat menyelesaikan pekerjaan dalam suatu satuan waktu daripada CISC yang lebih sedikit.
- *Benchmark* sintetis. Metoda–metoda bench-marking yang dikembangkan antara lain Whetstone, Drystone, Linpack, NAS Parallel Benchmark (NPB), Perfect Club, iCOMP, SLALOM, Peak FLOPS, STREAM, peak memory bandwidth, LLL (Lawrence Livermore Loops), HINT (*Hierarchial INTegration*), SPEC. Pengukuran dengan *benchmark* memiliki dua kelemahan yaitu keragu-raguan apakah program yang disusun cukup mewakili keadaan yang sebenarnya atau tidak dan mudahnya manipulasi terhadap hasil perhitungan akibat optimasi yang dilakukan pada kompilator.

2. Unjuk Kerja Sistem Komputer

Istilah performansi atau unjuk kerja ditujukkan kepada layanan yang diberikan oleh orang atau mesin. Kaitannya dengan sistem informasi yang dihasilkan oleh sistem komputer, unjuk kerja menunjukkan kemampuan yang diberikan oleh seluruh fasilitas dalam menyediakan informasi kepada pemakai. Fasilitas di sini mencakup perangkat keras dan perangkat lunak, misalnya bahasa pemrograman yang akan melakukan komunikasi dengan sistem, alat bantu yang digunakan dalam mendesain dan mengembangkan program, mekanisme proses dan penyembuhan kesalahan (*fault recovery*), tingkat pengamanan data, dan lain-lain.

Pemilihan sistem komputer tergantung pada beberapa faktor, diantaranya yaitu unjuk kerja yang ditentukan oleh tipe pemakai dan tipe aplikasi yang digunakan. Indek unjuk kerja dapat dievaluasi dengan berbagai cara, misalnya dengan pengukuran, pengalkulasian atau perkiraan.

Untuk sistem komputer processor tunggal, pengukuran unjuk kerja dapat dilakukan dengan dua

ISBN: 979-756-061-6

teknik diatas yaitu laju detak (*clock*) processor dan *benchmarking* sintetis.

2.1 Benchmarking

Benchmarking merupakan salah satu teknik yang dapat digunakan untuk mengukur unjuk kerja sistem komputer keseluruhan atau masing-masing komponennya, seperti CPU, RAM, Video Card, Hard disk atau jaringan..

Beban kerja yang diberikan pada sistem komputer saat pengujian, dapat memilih salah satu cara yaitu beban kerja nyata atau beban kerja sintetis.

Beban nyata dilakukan dengan mengukur unjuk kerja komputer pada saat menjalankan program yang sebenarnya, misalnya pada proses pembuatan halaman web, aplikasi yang dijalankan antara lain Adobe Photoshop, Macromedia DreamWeaver, dll. Proses pengukuran akan memakan waktu yang agak lama sesuai dengan kemampuan sistem komputer menjalankan beberapa program secara bersamaan.

Berbeda dengan beban kerja nyata, pada beban kerja sintetis dilakukan dengan membuat program khusus yang akan digunakan untuk mengukur unjuk kerja sistem komputer. Waktu pengukuran yang diperlukan sangat cepat. Namun, akurasi dari pengukuran dengan teknik masih banyak yang meragukan.

Metoda-metoda yang telah dikembangkan dan digunakan pada berbagai perangkat lunak pengujian, antara lain:

a. Whetstone

Whetstone merupakan *benchmark* sintetik yang dikembangkan oleh Curnow dan Wichman. *Benchmark* ini dimaksudkan untuk mengukur kinerja komputer dalam mengolah bilangan *floating point* dan digunakan untuk membandingkan arsitektur maupun kompilator teroptimisasi yang dijalankannya. (1) Program semula dibuat dalam bahasa Algol dengan kompilator Algol 60 yang menterjemahkannya menjadi instruksi untuk mesin Whetstone imajiner (Sill, 1996). Kelemahan *benchmark* Whetstone adalah kecilnya ukuran modul/program benchmark sehingga sistem memori di luar *cache* tidak teruji, dan dengan optimisasi kompilator dengan mudah didapatkan skor *benchmark* tinggi tanpa mengubah sistem yang diuji.

b. Dhrystone

Dhrystone juga merupakan *benchmark* sintetik yang dikembangkan oleh Reinhold Weicker pada awal tahun 1980-an dan difokuskan untuk mengukur kinerja komputer atas bilangan integer dan string. (13) Program asli ditulis dalam bahasa Ada, dan kemudian diterjemahkan ke dalam bahasa-bahasa lain. Sama seperti Whetstone, program *benchmark* Dhrystone memiliki ukuran yang terlalu kecil (sekitar 1,5 KB) sehingga tidak dapat menguji sistem di luar *cache*. Optimisasi kompilator juga dapat dilakukan untuk mempertinggi skor perolehan. (9)

c. Linpack

Linpack yang dikembangkan oleh Jack Dongarra, kernelnya dikembangkan dari rutin program aplikasi aljabar linier. Semula ditulis dan digunakan dalam lingkungan bahasa program Fortran namun tersedia juga versi bahasa C. Sebagian besar waktu uji merupakan waktu eksekusi subrutin yang menjalankan operasi matriks y(i) = y(i) + a * x(i). Versi standar bekerja dengan matriks ukuran 100x100, tetapi juga tersedia versi dengan ukuran matriks 300x300 dan 1000x100 dengan aturan optimisasi yang berbeda. Kelemahannya, program hanya mewakili tipe komputasi matriks yang memang banyak digunakan dalam bidang bidang sains. (4)

d. NAS Parallel Benchmark (NPB)

NAS Parallel Benchmark (NPB), yang dikembangkan oleh peneliti di NAS, yakni cabang dari NASA Ames Research Lab, untuk mengukur kinerja komputer paralel. NPB dikembangkan khusus untuk mengukur kinerja komputer paralel, yang memerlukan penulisan ulang program agar dapat secara efektif dan efisien membagi beban komputasi di antara prosesor-prosesornya. (2)

e. Perfect Club

Para peneliti di Universitas Illinois, yang telah lama bekerja dengan superkomputer, tidak puas dengan berbagai teknik yang dipakai dalam mengembangkan berbagai benchmark. Mereka mengembangkan benchmark yang disebut Perfect Club, yang memiliki pendekatan pengukuran mirip dengan SPEC. Perfect Club merupakan gabungan dari aplikasi-aplikasi riil yang disumbangkan oleh kelompok-kelompok peminat komputasi dan diorganisasi sedemikian rupa sehingga menjadi satu benchmark. Benchmark Perfect Club terutama mengukur kinerja atas bilangan floating-point dan biasanya dieksekusi pada superkomputer. Tujuan utama proyek Perfect Club adalah mengkarakterisasi program aplikasi dalam hal perilaku algoritmanya sehingga memungkinkan pemakai memperoleh prediksi kinerja yang diharapkan untuk program aplikasi yang.

f. iCOMP

Benchmark iCOMP dikembangkan oleh Intel untuk membandingkan kinerja prosesor-prosesor yang ada di pasaran. Ketika prosesor generasi 486 diperkenalkan, di pasaran muncul berbagai versi mulai 486SX, 486DX2, dan sebagainya. Agar calon pembeli memiliki gambaran ringkas kinerja prosesor, Intel mengembangan angka indeks yang

ISBN: 979-756-061-6

merupakan angka kinerja prosesor dibandingkan dengan kinerja prosesor rujukan. Benchmark ini khusus mengukur kinerja prosesor dan tidak mencerminkan kinerja komputer secara keseluruhan. Popularitas iCOMP terutama karena benchmark ini hampir selalu menjadi ukuran kinerja prosesor-prosesor Intel dalam iklan-iklannya (setidaknya sampai generasi Pentium).

g. Slalom

Sebagai upaya untuk mengembangkan *benchmark* yang lebih baik, peneliti di Ames Laboratory merancang program yang disebut sebagai SLALOM (*The Scalable, Language-independent, Ames Laboratory One-minute Measurement*). *Benchmark* SLALOM mengukur kinerja komputer dengan pendekatan waktu-tetap (*fixed-time*), bukan ukuran-masalah tetap (*fixed-size problem*). Dengan prinsip waktu-tetap, dimungkinkan membandingkan berbagai jenis komputer, dari komputer personal sampai komputer paralel berkemampuan besar. SLALOM, yang secara otomatis menyesuaikan dengan daya komputasi (*computing power*) yang ada dan memperbaiki kekurangan berbagai *benchmark* sebelumnya, memiliki sifat-sifat: sangat terskala, memecahkan persoalan riil, memperhitungkan juga waktu yang diperlukan untuk unit masukan dan keluaran, dan dapat dijalankan pada komputer paralel serta menggunakan berbagai bahasa yang ada.

Meskipun dirancang untuk memanfaatkan memori sesuai dengan kecepatan komputer, SLALOM tidak dapat dijalankan selama satu menit (waktu yang diperlukan untuk pengukuran) pada komputer yang kapasitas memorinya relatif kurang terhadap kecepatannya. Akibatnya, komputer dengan kapasitas memori kecil tidak dapat diukur kinerjanya dengan SLALOM.

h. Peak FLOPS

Peak FLOPS, biasanya dengan menggambarkan hasil rating operasi pertambahan dan pembagian floating-point dari perangkat keras yang tidak terpengaruh oleh operasi lainnya.

i. STREAM, peak memory bandwidth

STREAM adalah kumpulan operasi looping yang sangat kecil. STREAM mencoba melakukan perkiraan total rate pada semua alamat memori yang dapat mengirimkan data ke processor tanpa terganggu oleh operasi yang lain. *Peak,* mengukur efek penyegaran memori, interupsi input/output, atau beban pada bus memori, dengan mengabaikan faktor pengali lebarnya bus. (9)

j. LLL (Lawrence Livermore Loops)

Lawrence Livermore Loops, didesain oleh Frank McMahon dengan mengutip program aplikasi dalam Fortran yang digunakan pada Laboratorium Nasional Lawrence Livermore. (8)

k. HINT (*Hierarchial INTegration*)

Benchmark dengan pendekatan lain juga dibuat oleh peneliti di Ames Laboratory, dikenal dengan nama HINT (*Hierarchial INTegration*). Benchmark ini menghasilkan satuan ukuran yang disebut QUIPS (*Quality Improvement Per Second*) dan tidak menggunakan ukuran-masalah tetap ataupun waktu-tetap. HINT dikembangkan berdasarkan *benchmark* SLALOM tetapi bekerja lebih cepat. Bedanya, HINT tidak mematok ukuran masalah (*problem size*) maupun waktu komputasi. QUIPS merupakan satuan yang digunakan untuk mengukur banyaknya usaha yang dilakukan komputer pada rentang waktu tertentu. Gustafson, peneliti yang mengembangkan HINT, tidak menginginkan waktu yang terlalu singkat untuk melakukan pengukuran karena kebanyakan komputer bekerja sangat cepat pada awalnya, lalu mulai menurun kecepatannya setelah banyak terjadi kemelesetan (*miss*) *cache* dan mulai menggunakan memori utama atau bahkan harus mengakses data pada *harddisk.* (6, 7)

HINT diklaim sebagai *benchmark* yang memungkinkan pembandingan yang adil terhadap perbedaan-perbedaan ekstrim dalam hal arsitektur komputer, kineja absolut, kapasitas memori, dan taraf presisi komputasi. HINT merupakan perbaikan dari SLALOM dalam hal linieritas (kualitas penyelesaian masalah, pemakaian memori, dan banyaknya operasi, semuanya proporsional), mudah dikonversi ke komputer dengan arsitektur berbeda, serta menyatukan taraf presisi dan ukuran memori ke dalam satu kinerja. Sampai saat ini HINT masih belum banyak diuji dan belum cukup populer untuk digunakan sebagai tolok-ukur kinerja komputer universal.

l. Standards Performance Evaluation Corporation (SPEC)

Sebagai upaya untuk mendapatkan tolok-ukur baku agar dapat membandingkan kinerja berbagai sistem komputer, sekelompok perusahaan besar antara lain: DEC, Hewlett-Packard, IBM, Intel, dan Sun sepakat membentuk lembaga non-profit yang diberi nama System Performance Evaluation Corporation.. Lembaga ini ditugasi untuk mengembangkan dan memberi dukungan terhadap pembakuan *benchmark* kinerja komputer.

Sebelum membuat program untuk mengukur kinerja komputer, SPEC telah mempelajari sejumlah program yang umum dipakai, menganalisis algoritma dan bahasa mesinnya, menentukan cara mengukur kinerja komputer, dan menentukan rumusan untuk membuat rerata skor kinerja komputer dari skor-skor yang diperoleh masing-masing elemen *benchmark. Benchmark* SPEC terdiri atas dua kelompok program. Satu

ISBN: 979-756-061-6

kelompok merupakan program-program yang dititik-beratkan pada operasi atas bilangan *integer* dan satu kelompok lainnya dititikberatkan pada operasi atas bilangan *floating-point*. (5)

m. Fhourstonesnsieve, heapsort, Hanoi, queens, flops, fft, mm

Metoda *benchmark* yang menggunakan bilangan bulat dan floating-point (1).

2.2 Geometri Fraktal

Fraktal adalah suatu bentuk geometris yang dapat dipisahkan ke dalam bagian-bagian, dimana masing-masing bagian itu adalah versi kecil dari versi keseluruhannya. (11)

Kalau sebatang pohon bercabang, cabangnya berdahan, dahannya beranting, dan ranting itu mempunyai anak ranting yang lebih kecil, maka inilah fenomena fraktal. Evolusi sebuah fraktal biasanya chaotic. Benda fraktal hadir di sekitar kita, mulai dari skala mikro, makro, hingga mega dan giga. Percabangan akar, pola retakan batuan, daun cemara, bahkan beberapa motif batik, adalah contoh-contoh fraktal skala makro. Kelokan-kelokan sungai, busur kepulauan, bentuk galaksi, nebula, adalah fraktal dalam skala mega hingga giga.

Geometri fraktal mempunyai karakter-karakter penting antara lain *self similar, self affine, self inverse,* dan *self squaring*. Yang jelas skala panjangnya tidak spesifik atau invariant. Berbeda dengan geometri euklidean, penyekalaan fraktal dicirikan oleh bilangan-bilangan pecahan atau tak bulat (*noninteger*), yang disebut dimensi fraktal (*fractal dimensions*). Ciri-ciri yang biasanya dijumpai pada bangun fraktal, adalah bahwa bagian terkecil dari benda itu merupakan cerminan bentuk keseluruhannya (the part is reminiscent of the whole). Dengan kata lain, bahwa di dalam suatu himpunan fraktal, bagian dari himpunan tersebut merupakan skala kecil dari keseluruhannya.

Geometri fraktal adalah salah satu cabang matematika yang mempelajari properti dan kelakuan fraktal. Geometri fraktal mengungkapkan berbagai situasi yang tidak dapat dengan mudah dilakukan oleh geometri klasik, dan telah banyak diaplikasikan pada berbagai bidang ilmu pengetahuan, teknologi dan komputer grafik.

2.2.1 Kategori Fraktal

Fraktal dapat dikelompokkan kedalam tiga kategori, yang menentukan cara pendefinisian atau generasi fraktal tersebut yaitu:

a. Fungsi sistem iterasi. Contohnya fraktal Cantor, permadani Sierpinski, Sierpinski gasket, kurva Peano, kepingan salju Koch, kurva ular naga Harter-Heighway, T-Square, spons Menger.
b. Fraktal yang didefinisikan oleh hubungan kambuh (*recurrence*) pada setiap titik. Contohnya fraktal Mandelbrot dan Lyapunov.
c. Fraktal acak, yang digenerasi oleh stokastik. Contohnya fractal landscape dan Levy flight.

Fraktal dapat dikelompokkan juga berdasarkan konsep *self-similarity*, yaitu:

a. *Exact self-similarit*y
   Merupakan jenis *self-similarity* yang paling kuat, fraktal nampak serupa pada berbagai skala. Fraktal ditentukan oleh fungsi sistem iterasi sering menampilkan *exact self-similarity.*
b. *Quasi-self-similarity*
   Merupakan bentuk *self-similarity* lepas, fraktal nampak seolah-olah sama (namun tidak persis sekali) identik pada berbagai skala. Fraktal *quasi-self-similarity* terdiri dari salinan kecil-kecil semua fraktal yang didisimpangkan dan bentuk penurunan. Fraktal hubungan kambuh (*recurrence*) biasanya menggunakan *quasi-self-simiarity*.
c. *Statistical-self-similarity*
   Merupakan tipe self-similarity yang paling lemah, fraktal memiliki ukuran kwantitatif atau statistik yang dipertahankan pada berbagai skala. Fraktal acak adalah contoh fraktal *statistical-self-similarity*.

Perlu diingat bahwa tidak semua obyek self-similarity adalah fraktal, contohnya garis Euclidian.

3. *Benchmarking* Sistem Komputer dengan Beban Kerja Fraktal

Pengujian sistem komputer yang dilakukan dengan memberikan beban kerja fraktal Mandelbrot. Komponen-komponen yang diuji antara lain processor, RAM, dan video card. Hasil pengujian berupa jumlah iterasi yang dapat dalam satuan waktu detik.

Berikut ini contoh hasil pengukuran:

```
Number of CPU: 1
SSE instructions supported. Switching to
SSE quadpoints computation.
SSE2 instructions supported.
3DNow! instructions NOT available.
Thread 1 says: "I'm a slave, I'm alive."
OpenGL v1.4.0
  Renderer:   GeForce4   MX   440   with
AGP8X/AGP/SSE2
  Vendor: NVIDIA Corporation
BENCHMARK (Using 1 CPU, no render)
 Sys perf timer freq: 3579545 Hz
 size:  500*500
 maxiters:      9999
 rangex:        -2.00 to 1.00
 rangey:        -1.50 to 1.50
  SSE benchmark:
    1.009 sec
    417.709 MegaIters/sec
  SSE2 benchmark:
    1.989 sec
    211.847 MegaIters/sec
```

ISBN: 979-756-061-6

```
  3DNow! benchmark:
    Not supported.
  FPU ASM benchmark:
    4.041 sec
    104.046 MegaIters/sec
  FPU C benchmark:
    12.793 sec
    32.865 MegaIters/sec
  GPU VertexProgram benchmark (beta!
maxiters=10) on GeForce4 MX 440 with
AGP8X/
AGP/SSE2:
    7.208 sec
    69.364 MegaIters/sec
  GPU FragmentProgram benchmark (beta!
maxiters=20) on GeForce4 MX 440 with
AGP8
X/AGP/SSE2:
ARB Fragment Program: Required
extensions are not supported Not
supported.
```

4. Kesimpulan

Benchmarking sistem komputer menjadi salah satu kebutuhan bagi pengguna komputer yang memerlukan unjuk kerja sistem yang tetap prima. Kegiatan ini bisa dikerjakan tanpa bantuan dari seorang tenaga ahli, karena telah banyak perangkat lunak bebas yang dapat dijumpai dipasaran. Pemakaian geometri fraktal pada pemberian beban kerja terhadap komputer pada *benchmarking*, menjadi pilihan yang tepat dalam menguji sistem komputer. Metoda pengujian terkadang cepat usang dikarenakan tidak relevan lagi dibandingkan dengan teknologi hardware yang terus mengalami perubahan kemampuan sangat cepat. Metoda yang dipakai diharapkan dapat menjawab permasalahan ini karena parameter pada fraktal bisa diset disesuaikan dengan kebutuhan.

Daftar Pustaka

[1] A. Aburto Jr., *benchmarks developed and maintained at the Naval Command Control and Ocean Surveillance Center RDTE Division (NRaD), San Diego, CA*. ftp site: ftp.nosc.mil in directory pub/aburto. Mirrored at the NIST Web site.

[2] D. Bailey, J. Barton, T. Lasinski, and H. Simon, *The NAS Parallel Benchmarks*, *ReportRNR-91-002*, NASA/Ames Research Center, January 1991.

[3] Curnow, H. J., and Wichmann, B. A., *A Synthetic Benchmark*, *Computer Journal*, 19,1, February 1976.

[4] J. Dongarra, *Performance of Various Computers Using Standard Linear Equations Software in a Fortran Environment*, ORNL, updated periodically.

[5] J. Dongarra, W. Gentzsch, eds, *Computer Benchmarks*, North Holland, Amsterdam, 1993.

[6] J. Gustafson and Q. Snell, *HINT: A New Way to Measure Computer Performance*, *Proceedings of the Twenty-Eight Hawaii International Conference on System Sciences*,Wailea, Maui, Hawaii, January 1995.

[7] J. Gustafson, Q. Snell, R. Todi, HINT Web site: http://www.scl.ameslab.gov/HINT

[8] McMahon, F.H., *L.L.N.L FORTRAN KERNELS: MFLOPS*, Lawrence Livermore National Laboratory, (benchmark tape available), 1983.

[9] The STREAM Web site is http://reality.sgi.com/mccalpin/papers/bandwidth/bandwidth.html

[10] R. Weicker, *Dhrystone: A Synthetic Systems Programming Benchmark*, *Communications of theACM*, Volume 27, Number 10, October 1984.

[11] Roger T. Stevens, *Fractal programming in C*, M&T Publishing Inc, USA, 1989.