ANALISIS KINERJA BEBERAPA ALGORITMA LOAD BALANCING

Helmi Kurniawan[1)], Reza Pulungan [2)]
[1]*Program Studi Teknik Informatika STMIK Potensi Utama*
[2]*Jurusan Ilmu Komputer dan Elektronika, Universitas Gadjah Mada, Yogyakarta*
E-mail: helmikk12@gmail.com, pulungan@ugm.ac.id

Abstrak
Load balancing adalah proses peningkatan kinerja paralel dan sistem terdistribusi melalui redistribusi beban antara prosesor [1] [5]. Dalam tulisan ini penulis menyajikan analisis kinerja berbagai algoritma load balancing berdasarkan parameter yang berbeda, mempertimbangkan dua load balancing khas pendekatan statis dan dinamis. Hasil analisis menunjukkan bahwa statis dan dinamis kedua jenis algoritma dapat memiliki kemajuan serta kelemahan satu sama lain. Memutuskan jenis algoritma yang akan dilaksanakan akan didasarkan pada jenis aplikasi paralel untuk memecahkan. Tujuan utama dari makalah ini adalah untuk membantu dalam desain algoritma baru di masa depan dengan mempelajari perilaku berbagai algoritma yang ada.

***Kata kunci**:load balancing (LB), beban kerja, sistem terdistribusi, Load balancing Statis, Dinamis Load Balancing*

1. PENDAHULUAN

Dalam sistem paralel dan mendistribusikan lebih dari satu prosesor memproses program paralel. Jumlah waktu proses yang diperlukan untuk melaksanakan semua proses ditugaskan untuk prosesor disebut beban kerja prosesor. Sebuah sistem [2] [3] komputer didistribusikan dengan puluhan atau ratusan komputer yang terhubung dengan jaringan kecepatan tinggi memiliki banyak keuntungan dibandingkan sistem yang memiliki komputer standalone yang sama. Sistem terdistribusi menyediakan berbagi sumber daya sebagai salah satu keuntungan utama, yang menyediakan kinerja yang lebih baik dan keandalan daripada sistem tradisional lain dalam kondisi yang sama. Salah satu isu penelitian secara paralel dan sistem terdistribusi adalah pengembangan teknik yang efektif penyebaran beban kerja pada beberapa prosesor. Tujuan utamanya adalah untuk mendistribusikan pekerjaan antar prosesor untuk memaksimalkan throughput, menjaga stabilitas, pemanfaatan sumber daya dan harus fault tolerant di alam.

Lokal penjadwalan yang dilakukan oleh sistem operasi terdiri dari distribusi proses untuk irisan waktu prosesor. Di sisi lain penjadwalan global adalah proses memutuskan dimana untuk melaksanakan proses dalam sistem multiprosesor. Penjadwalan global dapat dilakukan oleh elemen pemrosesan tunggal pusat atau master, atau boleh didistribusikan antar elemen pengolahan.

Global penjadwalan lebih lanjut diklasifikasikan ke dalam kategori penjadwalan statis dan dinamis. Dalam proses penjadwalan statis ditugaskan untuk prosesor sebelum eksekusi dimulai. Di sisi lain penjadwalan dinamis dapat menetapkan kembali proses untuk prosesor selama eksekusi. Load sharing dan load balancing adalah klasifikasi lebih lanjut penjadwalan dinamis.

Beban perjuangan berbagi untuk menghindari negara unshared di prosesor yang tetap diam sementara tugas bersaing untuk layanan di beberapa prosesor lain. Load balancing juga melakukan hal yang sama namun ia pergi satu langkah di depan untuk berbagi beban dengan mencoba untuk menyamakan beban pada semua prosesor. Load balancing adalah untuk memastikan bahwa setiap prosesor pada sistem tidak kira-kira jumlah pekerjaan yang sama pada setiap titik waktu. Proses dapat bermigrasi dari satu node ke yang lain bahkan di tengah pelaksanaan untuk memastikan beban kerja sama. Algoritma untuk load balancing harus bergantung pada asumsi bahwa informasi tangan di masing node akurat untuk mencegah proses dari secara terus menerus beredar tentang sistem tanpa kemajuan.

Load balancing adalah salah satu prasyarat untuk memanfaatkan sumber daya penuh paralel dan sistem terdistribusi. Load balancing mungkin terpusat di prosesor tunggal atau didistribusikan di antara semua elemen pemrosesan yang berpartisipasi dalam proses load balancing. Beberapa tugas yang dijadwalkan untuk prosesor yang terpisah, berdasarkan beban CPU saat ini pada masing-masing. Banyak penelitian telah dilakukan pada load balancing untuk bertahun-tahun dengan tujuannya adalah untuk menemukan skema load balancing dengan overhead serendah mungkin.

2. ISI MAKALAH

Dalam tulisan ini, telah melakukan studi dari enam algoritma load balancing, berbagai parameter yang digunakan untuk memeriksa hasilnya. Makalah ini terdiri dari I Pendahuluan berisikan pengantar singkat III algoritma load balancing statis, IV memberikan pengenalan algoritma load balancing yang dinamis, V memberikan parameter untuk menganalisis algoritma VI hasil studi dengan bantuan tabel I dan VII kesimpulan.

3. STATIS LOAD BALANCING

Dalam metode ini kinerja [3] [6] dari prosesor ditentukan pada awal eksekusi. Kemudian tergantung pada kinerja beban kerja didistribusikan dalam memulai oleh master prosesor. Prosesor budak menghitung pekerjaan mereka dialokasikan dan menyampaikan hasil mereka untuk menguasai. Sebuah tugas selalu dijalankan pada prosesor untuk yang diberikan yang adalah metode beban statis balancing adalah non-preemptive. Tujuan metode load balancing statis adalah untuk mengurangi waktu pelaksanaan keseluruhan program bersamaan sambil meminimalkan penundaan komunikasi. Kelemahan umum dari semua skema statis adalah bahwa pilihan akhir dari sebuah host untuk proses alokasi dilakukan saat proses dibuat dan tidak dapat diubah selama eksekusi proses untuk membuat perubahan pada load sistem.

A. Round Robin and Random Algorithms

Pada bagian robin [5] proses dibagi secara merata antara semua prosesor. Setiap proses baru yang ditugaskan untuk prosesor baru untuk putaran robin. Urutan proses alokasi dipertahankan pada setiap prosesor lokal independen dari alokasi dari prosesor. Dengan algoritma round robin sama beban kerja diharapkan untuk bekerja dengan baik. Round Robin dan skema Acak [6] bekerja dengan baik dengan jumlah proses lebih besar dari jumlah prosesor

Keuntungan dari algoritma Round Robin adalah bahwa hal itu tidak memerlukan komunikasi antar-proses. Round Robin dan algoritma Acak keduanya bisa mencapai kinerja terbaik di antara semua algoritma load balancing untuk aplikasi tertentu tujuan khusus. Dalam Robin Round umum dan Acak tidak diharapkan untuk mencapai kinerja yang baik dalam kasus umum.

B. Algorithm Middle Manager

Dalam algoritma ini [10], prosesor pusat memilih host untuk proses baru. Prosesor minimal dimuat tergantung pada beban keseluruhan yang dipilih ketika proses dibuat. Load manajer memilih host untuk proses baru sehingga beban prosesor menegaskan ke tingkat yang sama sebanyak mungkin. Dari informasi tangan di manajer sistem pembebanan beban negara pusat membuat load balancing penghakiman. Informasi ini diperbarui oleh prosesor remote, yang mengirim pesan setiap kali beban pada mereka perubahan. Informasi ini dapat bergantung pada menunggu proses induk penyelesaian proses anak-anaknya, akhir eksekusi paralel Manajer load load balancing membuat keputusan berdasarkan informasi beban sistem, sehingga keputusan terbaik ketika proses dibuat. Tingginya tingkat komunikasi antar-proses yang bisa membuat negara bottleneck. Algoritma ini diharapkan untuk melakukan lebih baik daripada aplikasi paralel, terutama bila kegiatan dinamis diciptakan oleh host yang berbeda.

C. Threshold Algorithm

Menurut algoritma ini, proses ditugaskan segera setelah penciptaan ke host. Host untuk proses baru dipilih secara lokal tanpa mengirim pesan jarak jauh. Setiap prosesor menyimpan salinan pribadi dari beban sistem. Beban prosesor bisa mencirikan oleh salah satu dari tiga tingkatan: underloaded, medium dan Overloaded. Dua parameter ambang tunder dan Tupper dapat digunakan untuk menggambarkan level.

Under loaded - load < tunder
Medium - tunder ≤ load ≤ tupper
Overloaded - load > tupper

Pada awalnya, semua prosesor dianggap berada di bawah dimuat. Ketika keadaan beban prosesor melebihi batas tingkat beban, maka mengirimkan pesan tentang negara beban baru untuk semua prosesor terpencil, secara teratur memperbarui mereka untuk keadaan beban aktual dari seluruh sistem. Jika negara lokal tidak kelebihan beban maka proses dialokasikan secara lokal. Jika tidak, remote di bawah prosesor dimuat dipilih, dan jika tidak ada tuan rumah tersebut terjadi, prosesnya juga dialokasikan secara lokal. Algoritma Ambang batas memiliki proses komunikasi yang rendah antar dan sejumlah besar alokasi proses lokal. Kemudian menurunkan alokasi overhead proses remote dan overhead dari pengaksesan memori jauh, yang menyebabkan peningkatan kinerja. Kelemahan dari algoritma ini adalah bahwa semua proses dialokasikan secara lokal ketika semua prosesor terpencil kelebihan beban. Sebuah beban pada satu prosesor kelebihan beban bisa jauh lebih tinggi dari pada prosesor kelebihan beban lain, menyebabkan gangguan signifikan dalam load balancing, dan meningkatkan waktu eksekusi sebuah aplikasi.

4. DYNAMIC LOAD BALANCING

Ini berbeda dari algoritma statis di bahwa beban kerja didistribusikan antara prosesor pada saat runtime. Sang guru memberikan proses baru untuk para anak berdasarkan informasi baru yang dikumpulkan [2] [7]. Tidak seperti algoritma statis, algoritma dinamis mengalokasikan proses dinamis ketika salah satu prosesor

menjadi di bawah dimuat. Sebaliknya, mereka buffer dalam antrian pada host utama dan dialokasikan secara dinamis atas permintaan dari host remote.

A. Central Queue Algorithm

Central Queue Algorithm [12] bekerja pada prinsip distribusi dinamis. Ini pusat kegiatan baru dan permintaan yang tidak terpenuhi sebagai antrian FIFO pada host utama. Setiap aktivitas baru tiba di queue manager dimasukkan ke dalam antrian. Kemudian, setiap kali permintaan untuk kegiatan diterima oleh manajer antrian, ia bisa menghilangkan kegiatan pertama dari antrian dan mengirimkannya ke pemohon. Jika tidak ada kegiatan siap dalam antrian, permintaan buffer, sampai aktivitas baru tersedia. Jika aktivitas baru tiba di queue manager sementara ada terjawab permintaan dalam antrian, permintaan pertama yang akan dihapus dari antrian dan kegiatan baru yang ditugaskan untuk itu.

Ketika beban prosesor berada di bawah ambang batas, manajer beban lokal mengirim permintaan untuk aktivitas baru kepada manajer beban pusat. Manajer beban pusat menjawab permintaan segera jika suatu aktivitas siap ditemukan dalam antrian proses-permintaan, atau antrian permintaan sampai aktivitas baru tiba.

B. Local Queue Algorithm

Fitur utama dari algoritma ini [12] adalah dukungan proses migrasi dinamis. Ide dasar dari algoritma antrian lokal alokasi statis dari semua proses baru dengan proses migrasi yang diprakarsai oleh tuan rumah pada saat beban yang berada di bawah ambang batas, adalah parameter yang ditetapkan pengguna dari algoritma. Parameter yang mendefinisikan jumlah minimal proses siap manajer beban upaya untuk memberikan pada setiap prosesor.

Awalnya, proses-proses baru yang dibuat pada host utama dialokasikan pada semua host di bawah dimuat. Jumlah kegiatan paralel yang diciptakan oleh paralel pertama membangun pada host utama biasanya cukup untuk alokasi pada semua host remote. Sejak saat itu, semua proses dibuat pada host utama dan semua host lainnya dialokasikan secara lokal. Ketika tuan rumah yang mendapat di bawah dimuat, manajer beban lokal berusaha untuk mendapatkan beberapa proses dari host remote. Ini secara acak mengirimkan permintaan dengan jumlah proses siap lokal untuk manajer beban terpencil. Ketika seorang manajer beban menerima permintaan seperti itu, hal itu membandingkan nomor lokal proses siap dengan jumlah yang diterima. Jika yang pertama lebih besar dari yang terakhir, maka beberapa proses yang berjalan dipindahkan ke pemohon dan konfirmasi afirmatif dengan jumlah proses yang ditransfer dikembalikan.

5. PARAMETER

Kinerja algoritma load balancing berbagai diukur dengan parameter berikut.

A. Overload Rejection

Jika Load Balancing tidak tindakan overload mungkin diperlukan tambahan penolakan. Ketika situasi overload berakhir maka langkah-langkah overload pertama penolakan dihentikan. Setelah periode penjaga singkat Load Balancing juga ditutup.

B. Fault Toleran

Parameter ini memberikan bahwa algoritma dapat mentolerir kesalahan berliku-liku atau tidak. Hal ini memungkinkan algoritma untuk terus beroperasi dengan benar dalam hal kegagalan beberapa. Jika kinerja menurun algoritma, menurunkan sebanding dengan keseriusan kegagalan, bahkan kegagalan kecil dapat menyebabkan kegagalan total load balancing.

C. Forecasting Accuracy

Peramalan adalah derajat kesesuaian hasil dihitung ke nilai sebenarnya yang akan dihasilkan setelah eksekusi. Algoritma statis memberikan tingkat akurasi yang lebih dari algoritma yang dinamis seperti dalam asumsi yang paling bekas dibuat selama waktu kompilasi dan di kemudian ini dilakukan selama eksekusi.

D. Stabilitas

Stabilitas bisa dicirikan dalam hal penundaan dalam transfer informasi antara prosesor dan keuntungan dalam algoritma load balancing dengan mendapatkan performa yang lebih cepat dengan jumlah waktu tertentu.

E. Sentralisasi atau Desentralisasi

Skema terpusat menyimpan informasi global di node yang ditunjuk. Semua nodepengirim atau penerima mengakses node yang ditunjuk untuk menghitung jumlah beban-transfer dan juga untuk memeriksa bahwa tugas harus dikirim atau diterima dari. Dalamload balancing didistribusikan, setiap node menjalankan keseimbangan secara terpisah.Node menganggur dapat memperoleh beban selama runtime dari antrian global bersama proses.

F. Properties of Load Balancing Algorithm

Load balancing Statis memberikan beban ke node probalistik atau deterministiktanpa pertimbangan peristiwa runtime. Hal ini umumnya tidak mungkin untuk membuat prediksi kali kedatangan beban dan waktu pengolahan yang dibutuhkan untuk beban masa berikutnya. Di sisi lain, dalam beban dinamis menyeimbangkan distribusi beban dibuat selama run-time berdasarkan tarif pengolahan saat ini dan kondisi jaringan. Kebijakan DLB dapat menggunakan baik informasi lokal maupun global.

G. Cooperative

Parameter ini memberikan bahwa apakah prosesor berbagi informasi antara mereka dalam membuat keputusan alokasi proses lainnya tidak selama eksekusi. Apa parameter ini adalah tingkat kemerdekaan yang masing-masing prosesor telah di simpulkan bahwa bagaimana seharusnya itu bisa menggunakan sumber daya sendiri. Dalam situasi koperasi semua prosesor memiliki akuntabilitas untuk melaksanakan sebagian sendiri tugas penjadwalan, tetapi semua prosesor bekerja sama untuk mencapai tujuan efisiensi yang lebih baik. Dalam tindakan non-kooperatif prosesor individu sebagai entitas independen dan sampai pada keputusan tentang penggunaan sumberdaya mereka tanpa pengaruh dari keputusan mereka pada sisa sistem.

H. Proses Migrasi

Parameter Proses migrasi memberikan ketika melakukan sistem memutuskan untuk mengekspor sebuah proses? Ia memutuskan apakah akan membuat secara lokal atau membuat pada elemen pengolahan remote. Algoritma ini mampu untuk memutuskan bahwa itu harus melakukan perubahan distribusi beban selama pelaksanaan proses atau tidak.

I. Pemanfaatan Sumber Daya

Pemanfaatan sumber daya termasuk load balancing otomatis Sistem terdistribusi dapat memiliki jumlah tak terduga proses yang menuntut kekuatan pemrosesan yang lebih. Jika algoritma ini mampu memanfaatkan sumber daya, mereka dapat pindah ke bawah prosesor dimuat lebih efisien.

6. PERBANDINGAN

Perbandingan algoritma load balancing berbagai nama parameter yang berbeda ditampilkan dalam Table I [7].

Tabel 1. PARAMETER PERBANDINGAN ALGORITMA LOAD BALANCING

Parameter	Round Robin	Random	Local Queue	Central Queue	Central Manager	Threshold
Overload Rejection	No	No	Yes	Yes	No	Yes
Fault Tolerant	No	No	Yes	Yes	Yes	Yes
Forecasting Accuracy	More	More	Less	Less	More	More
Stability	Large	Large	Small	Small	Large	Large
Centralized Decentralized	D	D	D	C	C	D
Dynamic Static	S	S	Dy	Dy	S	S
Cooperative	No	No	Yes	Yes	Yes	Yes
Process Migration	No	No	Yes	No	No	No
Resource Utilization	Less	Less	More	Less	Less	Less

Sumber: **Y.Wang and R. Morris ,** ***Load balancing in distributed systems***

7. KESIMPULAN

Load balancing algoritma bekerja pada prinsip bahwa situasi di mana beban kerja diberikan, selama waktu kompilasi atau pada runtime. Perbandingan di atas menunjukkan bahwa algoritma beban statis balancing lebih stabil dibandingkan dengan dinamis dan juga kemudahan untuk memprediksi perilaku statis, tetapi pada saat yang sama algoritma terdistribusi yang dinamis selalu dianggap lebih baik daripada algoritma statis.

DAFTAR PUSTAKA

[1] G. R. Andrews, D. P. Dobkin, and P. J. Downey, "Distributed allocation with pools of servers," in ACM SIGACT-SIGOPS Symp. Principles of Distributed Computing, Aug. 1982, pp. 73-83.

[2] S. Malik, "Dynamic Load Balancing in a Network of Workstation", 95.515 Research Report, 19 November, 2000.

[3] Derek L. Eager, Edward D. Lazowska , John Zahorjan, "Adaptive load sharing in homogeneous distributed systems", IEEE Transactions on Software Engineering, v.12 n.5, p.662-675, May 1986.

[4] H.S. Stone, "Critical Load Factors in Two-Processor Distributed Systems," IEEE Trans. Software Eng., vol. 4, no. 3, May 1978.

[5] Zhong Xu, Rong Huang, "Performance Study of Load Balancing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.

[6] R. Motwani and P. Raghavan, "Randomized algorithms", ACM Computing Surveys (CSUR), 28(1):33-37, 1996

[7] Y.Wang and R. Morris, "Load balancing in distributed systems," IEEE Trans. Computing. C-34, no. 3, pp. 204-217, Mar. 1985.

[8] M. Zaki, W. Li, and S. Parthasarathy. "Customized dynamic load balancing for a network of workstations". Journal of Parallel and Distributed Computing: Special Issue on Performance Evaluation, Scheduling, and Fault Tolerance, June 1997.

[9] S.P. Dandamudi, "Sensitivity evaluation of dynamic load sharing in distributed systems", IEEE Concurrency 6 (3) (1998) 62-72.

[10] P. L. McEntire, J. G. O'Reilly, and R. E. Larson, Distributed Computing: Concepts and Implementations. New York: IEEE Press, 1984.

[11] L. Rudolph, M. Slivkin-Allalouf, E. Upfal. A Simple Load Balancing Scheme for Task Allocation in Parallel Machines. In Proceedings of the 3rd ACM Symposium on Parallel Algorithms and Architectures, pp. 237-245, July 1991.

[12] William Leinberger, George Karypis, Vipin Kumar, "Load Balancing Across Near-Homogeneous Multi-Resource Servers", 0-7695-0556- 2/00, 2000 IEEE.