

LECTURE NOTES

MOBI8001 – Mobile Technology & Cloud Computing

Topik 06 - Resource Allocation in Mobile Cloud Computing

LEARNING OUTCOMES

1. Peserta mampu menerapkan konsep mobile cloud computing dalam menyelesaikan masalah-masalah teknis di dunia nyata.
2. Peserta memiliki kemampuan dalam menganalisa arsitektur, platform, dan teknologi-teknologi pendukung dari mobile cloud computing.
3. Peserta mampu mengevaluasi kemajuan dan tantangan penelitian dari teknologi mobile cloud computing

OUTLINE MATERI :

1. Pendahuluan
2. Signifikansi Resource-Allocation pada Mobile Cloud Computing (MCC)
3. Strategi Resource-Allocation pada MCC
4. Resource-Allocation berbasis Semi-Markov Decision Process (SMDP) pada MCC
5. Task-Scheduling dengan Algoritma berbasis Aktivitas
6. Resource-Allocation Menggunakan Middleware
7. Tantangan Riset pada Resource-Allocation di MCC
8. Kesimpulan

ISI MATERI

A. Pendahuluan

Munculnya komputasi mobile dan komputasi cloud telah melahirkan konsep baru cloud computing mobile (MCC). Perangkat mobile memiliki batasan daya penyimpanan dan daya untuk pemrosesan. Di MCC, perangkat seluler diperkuat dengan meng-offload tugas dan datanya ke cloud. Cloud adalah kumpulan sumber daya yang kaya seperti memori, storage, daya pemrosesan, jaringan, server, database, dan aplikasi. Pengguna cloud menggunakan sumber daya ini dengan cara bayar sesuai keinginan Anda.

Saat user mengirim permintaan layanan ke cloud, penyedia cloud mengalokasikan sumber daya yang diinginkan ke user. Jadi, sangat penting bagi penyedia cloud untuk menggunakan strategi alokasi sumber daya yang bagus untuk menjaga kualitas layanan (QoS) cloud. Beberapa metode alokasi sumber daya, strategi, algoritma, dan middleware telah dikembangkan. Dalam bab ini, berbagai framework dan isu mengenai alokasi sumber daya untuk MCC akan dibahas.

B. Signifikansi Resource-Allocation pada Mobile Cloud Computing (MCC)

Hanya alokasi sumber daya yang tepat yang dapat meningkatkan perangkat seluler dan memenuhi permintaan pengguna. Dengan demikian, cloud dapat mempertahankan QoS-nya. Strategi alokasi resource yang optimal dapat menghindari situasi berikut:

1. Perebutan sumber daya (resource contention): Situasi ini muncul saat dua aplikasi mencoba mengakses resource yang sama pada saat bersamaan.
2. Kelangkaan sumber daya: Situasi ini muncul bila ada resource yang terbatas
3. Fragmentasi sumber daya: Situasi ini muncul saat sumber daya diisolasi. Meskipun

melimpah, resource tidak bisa dialokasikan ke aplikasi yang dibutuhkan.

4. Over-provisioning: Situasi ini muncul saat aplikasi menerima lebih banyak resource daripada yang diminta.
5. Under-provisioning: Situasi ini terjadi saat aplikasi diberikan resource yang lebih sedikit daripada permintaan .

C. Strategi Resource-Allocation pada MCC

Ada beberapa strategi resource-allocation seperti ditunjukkan oleh gambar 1:

1. SMDP-based resource allocation
2. Task-scheduling using ABC algorithm
3. Resource allocation using middleware
4. Energy-aware resource allocation
5. Entrop and FIFO based resource allocation
6. An auction mechanism for resource allocation

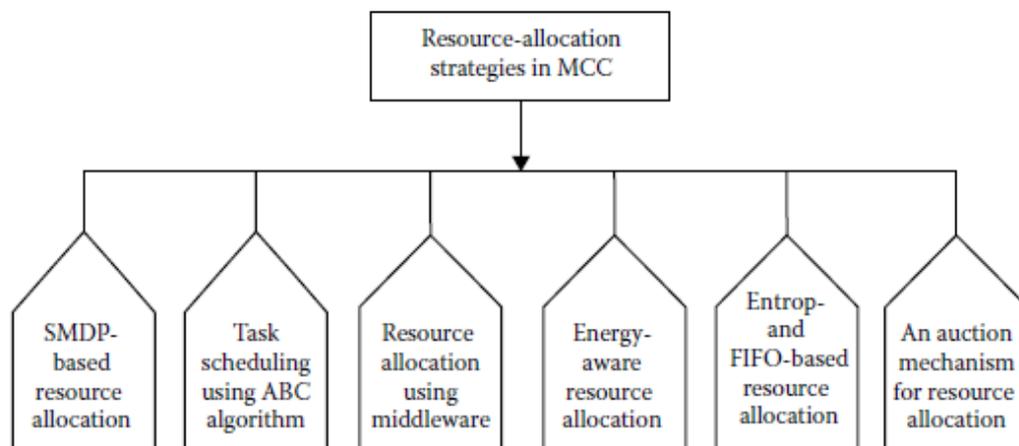


FIGURE 6.1
Resource-allocation strategies.

Gambar 1 Strategi Resource-Allocation

D. Resource-Allocation berbasis Semi-Markov Decision Process (SMDP) pada MCC

Cloud terdiri dari beberapa domain sesuai resource atau lokasi. Langkah-langkah yang harus diikuti untuk pemilihan dinamis domain cloud yang berdekatan adalah sebagai berikut:

Langkah 1: Menerima permintaan layanan dari klien.

Langkah 2: Tentukan apakah home domain bisa menampungnya atau tidak.

Langkah 3: Jika home domain bisa menampungnya, maka setuju permintaan layanan dan alokasikan resource untuk permintaan layanan tersebut.

Langkah 4: Jika home domain tidak dapat menampungnya, kirimkan permintaan transfer interdomain ke domain yang berdekatan.

Langkah 5: Kumpulkan keputusan dari interdomain yang berdekatan. Jika setidaknya satu domain yang berdekatan bisa menerimanya, maka pilih domain yang bisa menampungnya dan transfer layanan ke domain yang berdekatan yang dipilih. Jika domain yang berdekatan tidak dapat menerimanya, maka tolaklah permintaan layanan.

Saat permintaan layanan masuk ke cloud, pengontrol cloud mengalihkan permintaan ke home domain dari cloud sesuai dengan lokasi user. Kemudian, mesin virtual tunggal atau ganda (VMs) dialokasikan ke user sesuai permintaan mereka. Jika home domain tidak mencukupi untuk memenuhi permintaan user, maka permintaan tersebut akan ditransfer ke domain cloud lainnya.

Mengelola sumber daya awan di domain yang berbeda merupakan tugas yang penting. Pada paper Liang, et. al., sebuah sistem pengambilan keputusan layanan, dengan menggunakan SMDP untuk layanan interdomain yang mentransfer beban komputasi untuk menyeimbangkan beberapa domain awan, diusulkan. Teknik ini

meminimalkan jumlah penolakan layanan yang menurunkan tingkat kepuasan pengguna secara signifikan baik untuk sistem awan maupun pengguna dibandingkan dengan pendekatan serakah. Perbandingan dropping probability untuk interdomain transfer-request antara pendekatan SMDP dan greedy ditunjukkan oleh gambar 2.

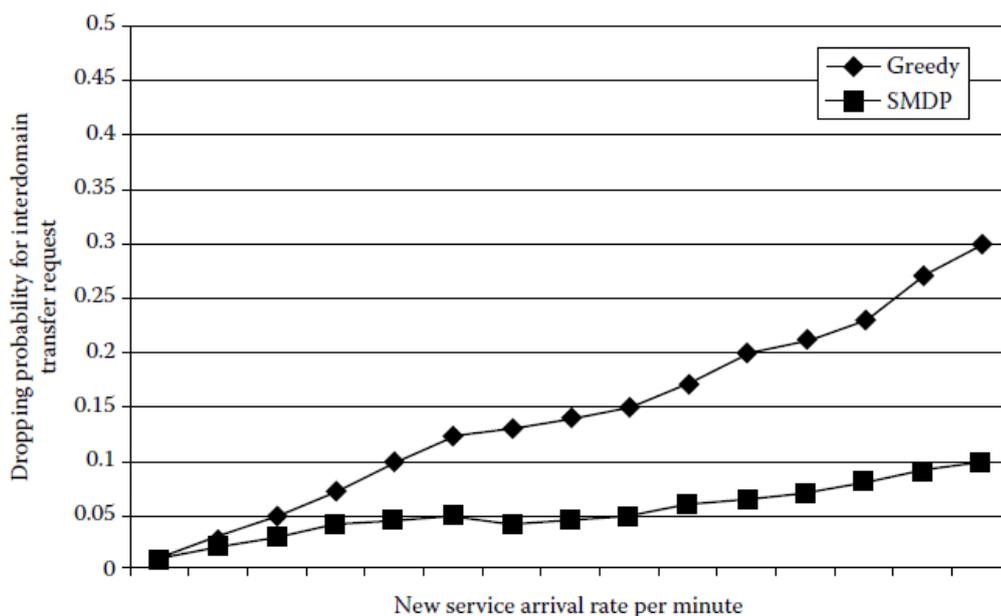


FIGURE 6.2
Comparison of SMDP and greedy approach on dropping probability for interdomain transfer request.

Gambar 2 Perbandingan Pendekatan SMDP dan Greedy

E. Task-Scheduling dengan Algoritma berbasis Aktivitas

Task-scheduling adalah proses di mana task yang di-offload dipetakan ke resource yang ada di cloud sesuai dengan persyaratan dan karakteristik task. Sing dan Ahmed [4] telah menggunakan algoritma activity-based costing (ABC) untuk membuat penjadwalan tugas menjadi optimal.

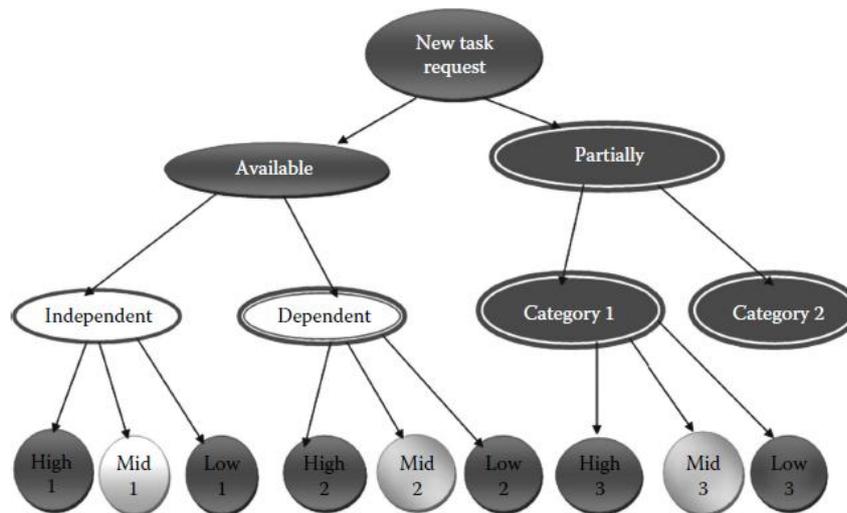


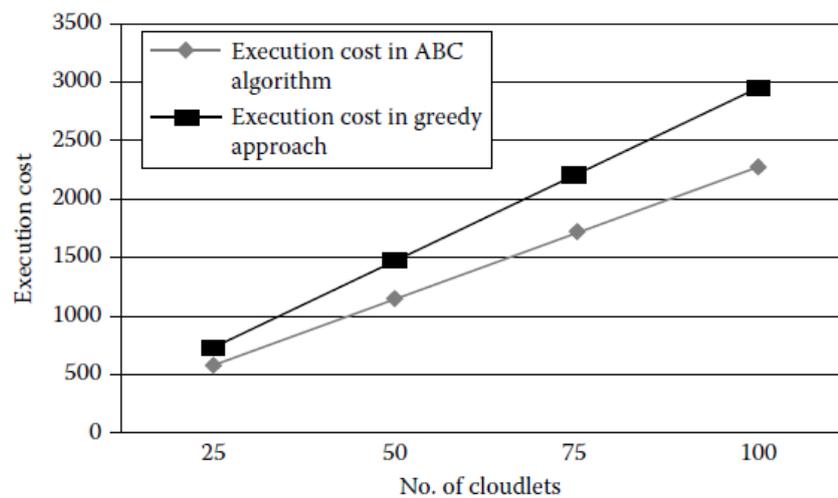
FIGURE 6.4
Activity-based algorithm for task scheduling.

Gambar 3 Algoritma berbasis Aktivitas untuk Task-Scheduling

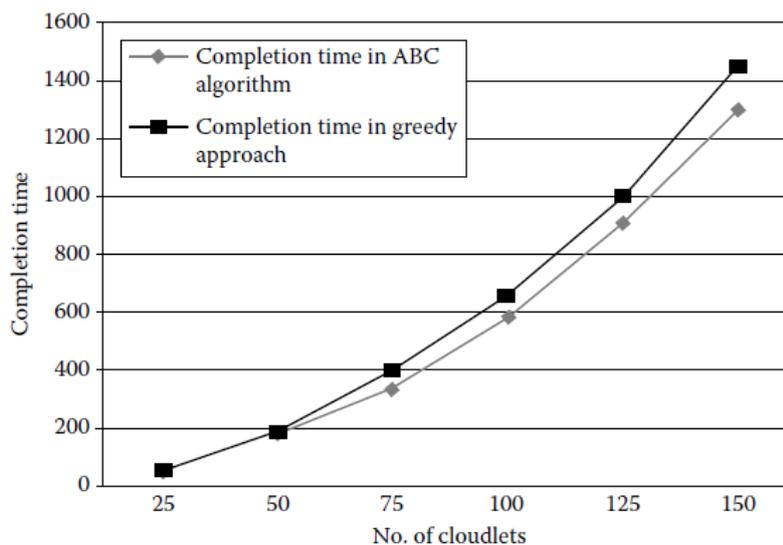
Algoritma ABC memiliki struktur hubungan parent-child atau struktur pohon seperti diilustrasikan oleh gambar 3. Setiap tugas memiliki waktu kedatangan yang spesifik. Menurut waktu kedatangan tugas, mereka disimpan dalam satu antrian parent. Setelah memeriksa resource dan data yang diminta untuk tugas tersebut, disimpan dalam dua antrian yang berbeda: antrian yang tersedia dan antrian yang tersedia sebagian. Tugas mungkin memiliki ketergantungan atau mungkin independen. Menurut ketergantungan tugas, mereka disimpan dalam antrian yang berbeda. Kemudian, tugas disimpan dalam antrian berbasis kategori yang berbeda jika mereka membutuhkan sumber data dari pusat data lainnya. Setelah antrian kategori, ada lagi tiga antrian berdasarkan prioritas tugas: tinggi, sedang, dan rendah. Prioritas diukur berdasarkan beberapa faktor seperti waktu penyelesaian, resource, ruang yang dibutuhkan, dan keuntungan.

Algoritma ABC efisien dalam biaya eksekusi dan waktu penyelesaian task. Sing

dan Ahmed [4] telah melakukan perbandingan dengan pendekatan alokasi sumber daya yang greedy dan telah membuktikan bahwa algoritma ABC berkinerja lebih baik ketika jumlah cloud [5] meningkat. Perbandingan kedua pendekatan ini dalam hal biaya eksekusi ditunjukkan pada Gambar 4. Perbandingan algoritma ABC dan pendekatan greedy dalam hal waktu penyelesaian tugas ditunjukkan pada Gambar 5.



Gambar 4 Pendekatan ABC vs greedy dalam biaya eksekusi



Gambar 5 Pendekatan ABC vs greedy dalam waktu penyelesaian task

F. Resource-Allocation Menggunakan Middleware

Di MCC, bagian perangkat seluler yang haus sumber daya di-offload ke cloud untuk lingkungan eksekusi yang lebih baik. Jenis aplikasi ini disebut aplikasi mobile cloud-assistive atau CAM-app. Ferber dkk. [6] telah mengusulkan sebuah middleware untuk alokasi resource untuk aplikasi CAM berbasis Java untuk memastikan permintaan user terhadap resource dari cloud. Metode ini mengimplementasikan keseluruhan fitur sisi server untuk memungkinkan CAM-apps. Fitur utamanya adalah sebagai berikut:

1. Hosting layanan remote
2. Pengelolaan dan alokasi sumber daya
3. Akuntansi dan penagihan

Arsitektur dari middleware yang digunakan ditampilkan oleh gambar 6.

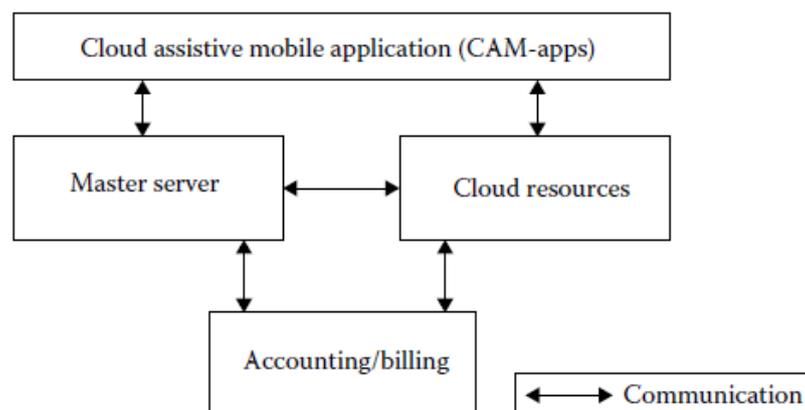


FIGURE 6.7
Architecture of the middleware.

Gambar 6 Arsitektur dari Middleware

G. Tantangan Riset pada Resource-Allocation di MCC

Ada beberapa tantangan riset untuk resource-allocation di MCC, antara lain sebagai berikut:

1. Energy-aware memory management

CPU dan memori adalah resource utama dari sistem cloud. CPU mengkonsumsi energi dengan jumlah tertinggi di cloud, dan memori adalah unit dengan daya tertinggi kedua. Hari demi hari, penggunaan cloud meningkat dengan cepat, dan memori dan CPU juga meningkat. Jadi, ini menghasilkan konsumsi daya yang sangat besar. Prosesor multi core sangat hemat energi. Di sisi lain, teknologi memori belum menunjukkan efisiensi energi. Pusat data memiliki masalah yang sama dalam penyimpanan jaringan dan disk.

2. Menjaga Service-Level Agreements (SLA) secara ketat.

SLA memainkan peran penting dalam menjaga QoS dari sistem cloud. SLA yang ketat selalu berusaha menghindari degradasi kinerja, yang merupakan pekerjaan yang sangat sulit. Banyak pertanyaan muncul di sini [9]: Bagaimana memprediksi performa? Bagaimana menentukan VMs mana, kapan, dan di mana harus dimigrasikan untuk mencegah degradasi kinerja jika sumber daya beberapa sistem dipertimbangkan? Bagaimana mengembangkan algoritma cepat dan efektif untuk optimasi penempatan VM di banyak sumber daya untuk sistem berskala besar? Pertanyaan-pertanyaan ini harus dijawab dalam riset-riset mengenai resource-allocation di masa mendatang.

3. Menyatukan berbagai strategi resource-allocation.

Banyak strategi resource allocation telah dikembangkan. Tapi setiap saat mereka tidak bisa mengoptimalkan setiap bidang. Misalnya, strategi resource allocation berbasis aktivitas [4] melakukan penjadwalan tugas dengan sangat baik, namun

tidak mempertimbangkan faktor efisiensi energi. Pendekatan penurunan best-fit yang dimodifikasi tidak mempertimbangkan faktor prioritas tugas [7]. Jadi dengan menggabungkan pendekatan yang berbeda, tradeoff positif antara kinerja dan efisiensi energi sistem dapat dilakukan.

SIMPULAN

Resource allocation merupakan salah satu masalah operasional utama di lingkungan MCC. Setelah tugas di-offload dari perangkat mobile ke dalam cloud, penyedia cloud mengalokasikan tugas dengan resource yang diinginkan. Dalam bab ini, kita membahas signifikansi, strategi, dan tantangan resource allocation masa depan dengan perbandingan pendekatan yang berbeda. Kita tentunya ingin mengubah kesulitan dari pendekatan sekarang dan mengatasi semua tantangan strategi alokasi sumber daya untuk mempertahankan QoS sistem MCC yang tepat.

DAFTAR PUSTAKA

1. DE, Debashis. Mobile cloud computing: architectures, algorithms and applications. CRC Press, 2016.
2. A. Berl, E. Gelenbe, M. D. Girolamo, G. Giuliani, H. D. Meer, M. Q. Dang, and K. Pentikousis, Energy-efficient cloud computing, *The Computer Journal*, 53(7), 1045–1051, 2010.
3. H. T. Dinh, C. Lee, D. Niyato, and P. Wang, A survey of mobile cloud computing: Architecture, applications, and approaches, *Wireless Communications and Mobile Computing*, 13(18), 1587–1611, 2013.
4. L. S. V. Sing and J. Ahmed, A greedy algorithm for task scheduling & resource allocation problems in cloud computing, *International Journal of Research & Development in Technology and Management Science*, 21(1), 1–17, 2014.
5. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, The case for VM-based cloudlets in mobile computing, *IEEE Pervasive Computing*, 8(4), 14–23, 2009.
6. M. Ferber, T. Rauber, M. H. C. Torres, and T. Holvoet, Resource allocation for cloud-assisted mobile applications, in *IEEE Fifth International Conference on Cloud Computing*, Honolulu, HI, pp. 400–407, 2012.
7. A. Beloglazov, J. Abawajy, and R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, *Future Generation Computer Systems*, 28(5), 755–768, 2012.