VOLUNTEER COMPUTING : PERUBAHAN DAN TANTANGAN DALAM VOLUNTEERCOMPUTING

Beny, S.Kom M.Sc
beny@stikom-db.ac.id
STIKOM DINAMIKA BANGSA
JAMBI

Abstrak

Volunteer computing telah memberikan dampak yang besar di komputasi terdistribusi modern. Cukup banyak proyek-proyek ilmiah diuntungkan dengan menggunakan konssep ini. Pada makalah ini penulis akan membahas mengenai sistem volunteer computing / komputasi sukarela dan sedikit melihat ke depan bagaimana perangkat-perangkat mobile bisa berpartisipasi dalam volunteer computing.

1. Pengantar

Volunteer computing memperlihatkan kekuatan dari orang-orang banyak. Bahkan volunteer computing pernah menjadi "komputasi berperforma tinggi" paling cepat di muka bumi, bahkan mengalahkan super komputer, "cluster", bahkan "grid" paling kuat sekalipun. Dengan sistem volunteer computing, pengguna bisa menyediakan sumber daya komputer mereka yang sedang menganggur untuk melakukan sesuatu yang berguna, yang biasana adalah projek ilmiah di universitas-universitas.

Dengan dukungan sumber daya yang begitu bervariasi, baik itu perangkat keras, perangkat lunak, kecepatan, ketersediaan, dan kehdanalan, hanya tinggal waktu saja dimana pemilik perangkat mobile modern seperti smart phone dan tablet dengan spesifikasi hardware yang tinggi mampu berbagi sumber daya mereka dan berpartisipasi dalam volunteer computing.

Pada makalah ini penulis akan membahas mengenai perkembangan volunteer computing dan mengulas bagaimana kemungkinan perangkat mobile digunakan layaknya PC dan laptop untuk menjadi bagian dalam lingkungan volunteer computing.

2. Sekilas mengenai Volunteer Computing / Komputasi Sukarela

Melakukan sesuatu secara sukarela adalah sesuatu yang datang dari kehidupan sosial manusia. Pekerja sukarela adalah seseorang yang bekerja untuk masyarakat tanpa meminta uang sebagai imbalan. Gagasan utama dari volunteer computing adalah mengajak pemilik komputer personal atau laptop untuk membagi sumber daya mereka (daya komputasi) untuk melakukan kegiatan-kegiatan yang berhubungan dengan ilmu pengetahuan, seperti menemukan bilangan prima, memecahkan enkripsi, mengkalkulasikan pelipatan protein, mencari keberadaan kehidupan di luar angkasa, mengkalkulasikan langkah-langkah dalam sebuah game (Wu I-Chen, et al., 2009), dan projek-projek lain yang dilaksanakan berdasarkan sistem ini.

Volunteer computing mempunyai hubungan yang rumit dengan grid computing, desktop grid computing, P2P, dan sistem terdistribusi secara umum. D.D Roure, et al. (2003), mengklasifikasikan evolusi dari grid computing menjadi tiga generasi, dimana generasi ke tiga lebih memilliki pdanangan holistik terhadap infrastruktur e-science yang mana hal ini sangat berkaitan erat dengan volunteer computing. Menurut Wang Yu et al. (2009) hingga sekarang sudah ada sistem volunteer computing yang telah diusulkan dalam jumlah besar dalam literatur-literatur yang menggunakan arsitektur yang berbeda-beda, mulai dari client-server hinga P2P, dan ini tentu memerlukan abstraksi yang lebih formal.

Yoshitomo Murata et al (2008) menyebutkan bahwa pendekatan volunteer computing bisa menyediakan kekuatan komputasi yang dibutuhkan untuk menjembatani jarak dalam skala waktu, tapi juga menimbulkan banyak masalah baru seperti keamanan, kehdanalan, dan koordinasi.

Kemampuan untuk menggunakan komputasi terdistribusi di luar lingkungan Universitas dan menjalankan projek yang melibatkan lebih dari seratus mesin dengan merekrut sukarelawan baru muncul setelah tahun 90-an. Berikut adalah projek-projek awal yang menerapkan system volunteer computing

a. GIMPS

Pada tahun 1996, projek volunteer computing berskala besar pertama adalah GIMPS (Great Internet Mersenne Prime Search) (http://mersenne.org). Projek ini bertujuan untuk menemukan bilangan prima.

Gambar 1. Logo GIMPS

Pada Januari 2013, project GIMPS telah berhasil menemukan bilangan prima terbesar, yakni $2^{57,885,161} - 1$ (Chris Caldwell, 2013).

b. Distributed.net

Pada tahun 1997, sebuah project bernama distributed.net diluncurkan dengan tujuan menembus enkripsi (http://distributed.net). Saat ini distributed. net sedang mengerjakan pemecahan enkripsi Rivest Chiper 5 (RC5) dengan 72 bits key.

Gambar 2. Logo Distributed.net

Beberapa projek yang berhasil dilakukan oleh distributed.net adalah:

- RSA Lab's 56-bit RC5 Encryption Challenge, selesai pada tanggal 19 October 1997 (setelah 250 hari dan 47% lingkup key diuji).
- RSA Lab's 56-bit DES-II-1 Encryption Challenge, selesai pada tanggal 24 Februari 1998 (setelah 39 hari)
- RSA Lab's 56-bit DES-II-2 Encryption Challenge, terhenti pada 17 Juli 1998 (ditemukan secara independen oleh cracker EFF DES setelah 2.5 hari)
- RSA Lab's 56-bit DES-III Encryption Challenge, Selesai 19 January 1999 (setelah 22.5 hours with the help of the EFF DES cracker)
- CS-Cipher Challenge, Selesai pada 16 Januari 2000 (setelah 60 hari dan 98% rentang kunci diuji).
- RSA Lab's 64-bit RC5 Encryption Challenge, Selesai pada 14 Juli 2002 (setelah 1757 hari dan rentang kunci diuji).

c. Proje-projek @home

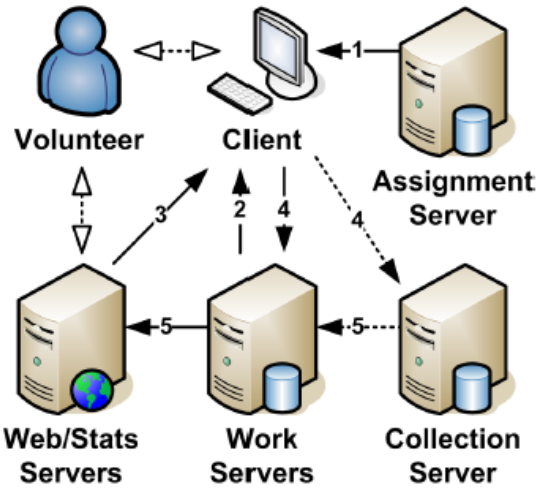
Pada tahun 1998, sebuah projek volunteer computing dengan skala lebih besar diluncurkan. SETI@home (Danerson, D.P., 2002) dan Folding@home (Pdane, V.S., 2002) mendapat liputan media yang lebih baik. Projek SETI@home berhubungan dengan pendeteksian kehidupan cerdas di luar bumi. Pendekatan pertama, dikenal dengan radio SETI, menggunakan radio teleskop untuk mendengarkan sinyal radio berpita sempit dari luar angkasa. Sinyal ini diketahui tidak terjadi secara alamiah, sehingga dengan terdeteksinya sinyal ini bisa menjadi bukti adanya teknologi luar angkasa. Sebelum muncul sebagai projek volunteer computing, projek SETI menggunakan supercomputer yang berlokasi di teleskop untuk mengerjakan analisa data dalam jumlah banyak. Hingga sekarang, projek SETI@home masih berjalan tanpa berhasil menemukan keberadaan mahluk luar angkasa, tapi ini membuktikan kepada komunitas ilmiah atas kekuatan volunteer computing.

Kisah sukses lain dari volunteer computing adalah projek Folding@home. Pemanfaatan utama dari infrasturktur Folding@home adalah kalkulasi statsitik terhadap lintasan molekul dinamis untuk model system biologi (Pdane, V.S., 2009). Dalam dinamika molecular, persamaan gerak Newton diintegrasikan untuk tiap atom pada system untuk memperbanyak lintasan.

Paralelisme terdistribusi pada Folding@home tidak digunakan untuk mempercepat kalkulasi individual, tapi digunakan untuk mendapatkan simulasi parallel dalam jumlah ratusan bahkan ribuan. Lintasan-lintasan ini kemudian membenarkan sebuah sampel statistis dari simulasi-simulasi, dimana di antaranya akan melipat (tergantung detail simulasi yang dilakukan) dan kebanyakan tidak melipat.

Folding@home menggunakan arsitektur client-server sebagaimana dapat dilihat pada Gambar 3 Arsitektur Folding@Home di bawah, dan cara dari system ini pada dasarnya mewakili system volunteer computing pada umumnya.

Yang membedakannya dengan komputasi terdistribusi lainnya, sebagai contoh komputasi grid, tugas-tugas diminta oleh client, dalam hal ini adalah PC atau laptop yang dimiliki oleh sukarelawan.



Gambar 3. Arsitektur Folding@home. Panah menunjukkan aliran data

Client akan meminta tugas dari server yang akan diberikan kepada Work Server. Setelah tugas diberikan kepada Work Server, langkah-langkah berikutnya adalah sebagai berikut:

1 Client akan menghubungi Work Server dan meminta unit kerja yang merupakan sekumpulan file input yang dibutuhkan client untuk dikerjakan dalam jangka waktu tertentu.

2 Berdasarkan unit kerja yang diberikan, client akan mengunduh inti komputasi yang dibutuhkan untuk mengerjakan tugas dari work server.

3 Ketika pekerjaan telah selesai, client akan mengirim hasilnya ke work server yang sama, atau ke server pengumpul yang telah ditentukan jika work server tidak dapat dijangkau.

4 Log files dan credits kemudian dikumpulkan dari work server dan diberikan kepada server statistic untuk tabulasi dan ditampilkan kepada partisipan. Statistik ini akan memacu para sukarelawan dari seluru dunia untuk "dipamerkan" atas apa yang telah mereka kerjakan. Untuk kegiatan yang lebih serius, para sukarelawan akan membentuk kelompok (berdasarkan Negara, daerah, dll) yang memungkinkan mereka menggabungkan nilai dari tiap anggota.

Sistem-sistem dari Volunteer computing banyak memanfaatkan dari CPU dan GPU multicore. Lingkungan baru seperti ini memerlukan algoritma yang harus dirancang ulang agar bisa bekerja. Prosesor multicore dari Nvidia atau ATi, dan IBM cell processor tentu akan mendongkrak performa dari projek Folding@home. Projek Folding@home telah mencapai 7 petaFLOPS pada Maret 2011 (Folding@home Wikipedia) . Sebagai pembandingan, supercomputer tercepat pada saat itu Tianhe-1 hanya berjalan pada 2.56 petaFLOPS (Tianhe-1 Wikipedia).

2.1 BOINC

BOINC (Berkeley Open Infrastructure for Network Computing) adalah platform popular yang mana banyak projek volunteer computing berbasis platform ini. BOINC adalah system perangkat lunak untuk ilmuwan yang dapat digunakan dan dioperasikan untuk projek komputasi berbasis sumber daya public (Danerson D.P., 2004). Dengan menggunakan BOINC, sebuah client (sukarelawan) bisa bergabung dengan lebih dari satu projek. Dengan kemampuan ini, sudah cukup banyak projek dibuat dan dijalankan pada platform BOIN ini. Beberapa projek yang telah dibahas sebelumnya, menggunakan BOINC sebagai basis sistemnya. Oleh karena BOINC telah digunakan oleh banyak projek, pada bagian ini akan dibahas secara mendalam mengenai mekanisme bagaimana platform ini bekerja untuk volunteer computing, dan bagaimana system ini mengatasi isu-isu dalam volunteer computing seperti keamanan, kehandalan dan koordinasi.

Untuk komputasi dan penanganan data, BOINC mengimplementasikan abstraksi yang sederhana tapi kaya untuk berkas-berkas, aplikasi dan data. Dan aplikasi dapat terdiri dari sekumpulan file-file sembarang. **Workunit (WU)** adalah representasi dari input untuk komputasi. Sebuah WU dapat terdiri dari sekumpulan referensi dari file input, sekumpulan argument baris perintah, dan variabel lingkungan. Tiap WU mempunyai parameter-parameter seperti perhitungan, memori dan kebutuhan media penyimpanan, serta batas waktu penyelesaian. File ini dibutuhkan oleh client untuk memulai pekerjaan. Setelah pekerjaan selesai hasil dalam bentuk file akan dikirim kembali ke server. Hasil dapat berupa referensi ke WU dan daftar referensi ke file-file output. File-file ini akan memiliki nama unik berdasarkan projek. File-file ini memiliki atribut-atribut yang menunjukkan indikasi tertentu, misalnya bahwa file itu harus tetap berada di sisi client setelah digunakan pertama kali, harus divalidasi menggunakan digital signature atau harus dikompresi sebelum dikirim melewati jaringan.

Untuk mengatasi hasil komputasi yang salah dari sisi client yang biasanya terjadi karena computer yang gagal fungsi, BOINC mempunyai mekanisme untuk mengatasi **redundant computing**. Sebuah projek dapat menentukan hasil N harus dibuat oleh setiap **working unit**. Jika M <= N dari hasil ini telah didistribusikan dan selesai, sebuah fungsi khusus akan dipanggil untuk membandingkan hasil dan mungkin memilih hasil yang sesuai aturan. Jika tidak ada kesepakan ditemukan, atau hasilnya gagal, BOINC akan membuat hasil baru untuk WU tersebut, dan melanjutkan proses sampai hasil maksimum tercapai atau batas waktu telah habis.

Dengan banyaknya sukarelawan yang ikut serta dalam volunteer computing, BOINC mempunyai beberapa mekanisme untuk mengatasi kondisi overload yang dapat menimbulkan

malapetaka yang disebabkan oleh koneksi yang bersamaan. Semua komunikasi client/server akan mundur secara eksponensial jika terjadi kegagalan. Sebaliknya, jika server BOINC hidup kembali setelah masa kritis berakhir, tingkat koneksi yang dapat dilakukan akan bersifat jangka panjang. Skema seperti ini juga diperluas untuk menangani kegagalan komputasi. Jika karena suatu sebab, sebuah aplikasi langsung gagal dalam sebuah host, client BOINC tidak akan menghubungi server secara berulang-ulang, melainkan akan menunda berdasarkan jumlah kegagalan.

Para sukarelawan akan diberikan pilihan yang cukup banyak yang membuat mereka merasa memiliki kendali atas mesin mereka di dalam system BOINC. Fitur seperti ini diperlukan karena partisipan hanyak akan berpartisipasi ke dalam volunteer computing hanya jika mereka merasa tidak begitu tingginya biaya, ataupun resiko untuk melakukannya. Dalam BOINC ada **general preferences** (preferensi umum) untuk user untuk menentukan batas hysteresis atas work buffering pada mesin-mesin, apakah BOINC tetap bisa bekerja ketika input dari mouse/keyboard sedang aktif, pada jam-jam apa saja BOINC melakukan kerjanya; berapa banyak ruangan cakram yang dapat BOINC gunakan; berapa banyak bandwidth jaringan yang dapat digunakan oleh BOINC; dan seterusnya.

Sebagaiaman telah dijelaskan secara singkat sebelumnya, untuk memotivasi para partisipan dalam melakukan tugasnya di kemudian hari, sebuah statistic pencapaian disediakan untuk mereka. Statistik ini didasarkan atas **credits**. Sebuah **Credit** adalah perhitungan kombinasi dari komputasi, media penyimpanan, dan transfer jaringan. Participant sangat termotivasi dari kredit, dan tertarik dalam membandingkat ranking mereka dengan orang lain.

Dilihat dari sisi heterogenitas mesin milik partisipan, BOINC memiliki mekanisme bernama **anonymous platform mechanism**. Ini akan berguna bagi partisipan untuk beberapa alas an. Dengan mekanisme ini, partisipan akan mampu mengkompilasi aplikasi client mereka sendiri, atau atas dasar mesin mereka tidak didukung oleh projek, atau partisipan ingin mengoptimalkan client mereka untuk arsitektur tertentu. Participan bisa mengunduh dan mengkompilasi kode sumber aplikasi (atau mendapatkan executables dari sumber pihak ke tiga), dan via sebuah file konfigurasi XML configuration. Ketika client berkomunikasi dengan server projek, ia akan mengindikasikan bahwa platformnya adalah anonym, dan menyediakan daftar dari versi aplikasi yang tersedia; yang kemudian tinggal diberikan workunit.

Fitur penting yang terakhir dalam BOINC adalah **Local Scheduling Policy**. Client bergantung pada kebijakan ini dalam kapan mendapatkan tugas dan dari projek apa, dan tugas apa yang dilakukan pada kondisi tertentu. Menggunakan kebijakan ini ada beberapa tujuan:

1 Memaksimalkan pengguanaan sumber daya (cth: untuk menjaga tiap prosesor tetap sibuk)

2 Untuk memenuhi batas waktu hasil

3 Untuk menjaga sumber daya partisipan untuk dibagikan ke projek-projek yang ada

4 Untuk menjaga variasi minimum di antara projek-projek

Di sisi lain, alokasi sumber daya pada server bergantung pada mekanisme apa yang digunakan (Liu Jun, et al., 2010).

3. Mobile Volunteer Computing

Ide untuk menggunakan perangkat keras computer mobile telah adalah selama beberapa decade. Sebelum adanya paradigma Volunteer Computing, ide atas mobile grid computing telah diajukan. Dikarenakan adanya hubungan dekat dan karakteristik antara grid computing dan volunteer computing, beberaa masalah dalam penggunaan perangkat mobile devices dalam grid selayaknya juga akan terjadi di volunteer computing. Tapi perkembangan perangkat keras ada perangkat mobile telah meningkat dengan sangat signifikan. Masalah yang menjadi isu utama pada grid sudah tidak lagi relevan (Litke A., 2004).

Untuk membuat definisi lebih jelas atas perangkat mobile, pada makalah ini penulis akan focus pada perangkat yang cukup banyak digunakan belakangan, seperti smart phones (Iphone, Danroid Phone, WP 7 Phone, dll) dan tablets (Ipad, Android Tablet, dll). Dengan meningkatnya jumlah perangkat mobile modern seharusnya dianggap sebagai sumber daya baru dalam volunteer computing.

Dengan menggunakan perangkat mobile seperti laptop atau notebook saat ini lebih layak dikategorikan sama dengan menggunakan PC desktop. Tapi sebagaimana dengan perangkat mobile modern, dikarenakan memiliki arsitektur CPU yang berbeda serta form factor yang berbeda, ada beberapa hal yang harus dipertimbangkan.

a. Hardware/Perangkat Keras

Keterbatasan kemampuan pemrosesan yang dahulu menjadi kendala dalam grid computing hanya memungkinkan perangkat mobile cocok menjadi resipien. Keterbatasan lain adalah media penyimpanan dan terbatasnya usia baterai, yang hingga sekarang baterailah yang cukup menjadi penghambat perangkat mobile dalam volunteer computing, tapi ini bisa diatasi dengan aturan-aturan tertentu seperti kapan proses pengerjaan tugas dilakukan.

Kemampuan pemrosesan smart phone sekarang telah melebihi 1.5 Ghz dengan delapan inti

processor (Arm Cortex-A9 Wikipedia). Ini juga akan ditemui pada perangkat tablet. Dengan kemampuan seperti itu, performa perangkat mobile ini sudah mulai mendekati computer personal desktop.

Sebagaimana telah disebutkan di bagian sebelumnya, system volunteer computing menghadapai masalah heterogenitas dikarenakan bervariasinya mesin yang dimiliki oleh partisipan. Tapi dengan arsitektur BOINC, masalah ini dapat diatasi dengan mekanisme **anonymous platform mechanism**, dan membawa perangkat mobile tentu akan lebih mudah. Permasalahan umur baterai dapat diatasi dengan cukup mudah dalam volunteer computing, dikarenakan partisipan dapat memilih dengan leluasa kapan dia ingin melakukan tugasnya, system dapat dipilih untuk hanya berjalan ketika perangkat sedang diisi listrik saja.

Yang perlu dipertimbangkan dengan lebih seksama adalah apakah perangkat keras mobile dirancang untuk memproses dalam jangka waktu tertentu? Desktop PC atau laptop dapat mengatasi panas yang dihasilkan dari komponen di dalamnya dikarenakan adanya kipas yang akan menyedot keluar panas sehingga aliran udara di dalam akan lebih dingin, tapi ini tidak ada pada perangkat mobile. Solusi paling realistis adalah membatasi waktu pengerjaan tugas untuk partisipan yang memiliki perangkat mobile ini dengan cara diberikan working unit yang kecil saja.

b. Sistem Operasi

Masih berkaitan dengan heterogenitas mesin dari partisipan, system operasi dari perangkat mobile juga bisa bekerja secara berbeda dalam menangani proses yang berjalan. Sistem operasi seperti Android akan mematikan proses yang dianggap terlalu memakan banyak sumber daya. Masalah stabilitas ini serupa ditemukan pada perangkat PC yang sering disebut sebagai masalah volatilitas, seperti aplikasi yang crasj, atau dimatikan oleh user. Untuk mengatasi masalah ini, system BOINC telah mengimplementasikan system ‘checkpoint’.

4. Boincoid

Membawa perangkat mobile ke dunia volunteer computing telah dimulai dari tahun 2008 oleh sekelompok mahasiswa dari Israel. Projek mereka adalah mem-porting client SETI@home ke lingkungan system operasi Android (http://boincoid.sourceforge.net) . Projek ini adalah opensource dan masih berjalan hingga sekarang.

Figure 4. Boincoid Logo

5. Kesimpulan

Volunteer computing telah menjadi solusi yang baik bagi ilmuwan untuk mendapatkan sumber daya komputasi dari public untuk projek ilmiah. Volunteer Computing pernah melewati supercomputer paling cepat pada masanya dan menjadi computer terdistribusi paling cepat. Bagi partisipan maupun projek, memiliki lebih banyak sumber daya akan semakin bagus. Dengan perkembangan perangkat keras yang sangat cepat untuk ponsel cerdas dan tablet, mempertimbangkan mereka untuk berpartisipasi akan membentuk paradigm baru dalam volunteer computing.

Daftar Pustaka

Danerson D.P, Cobb Jeff, Leboffski Matt, dan Werthimer Dan. SETI@home: Dan experiment in Public-Resource Computing. ACM, 2002.

Danerson D.P., BOINC: A System for Public-Resource Computing dan Storage, ACM, 2004.

Chris Caldwell, http://primes.utm.edu/largest.html, akses Maret 2013.

Liu Jun, Wan En Ze, Qiaou Jian Zhong, Lin Shu Kuan. An inframarginal analysis based resource allocation method in volunteer computing.Ninth International Symposium on Distributed Computing dan Applications to Business, Engineering dan Science, 2010.

Litke A., Skoutas D., Varvarigou T.,Mobile Grid Computing: Changes dan Challenges of Resource Management in a Mobile Grid Environment. PAKM, 2004.

Pdane V.S., Larson S.M., Snow C.D., dan M. Shirts. Folding@Home dan Genome@Home: Using distributed computing to tackle previously intractible problems in computational biology. Computational Genomics, Horizon Press, 2002.

Pdane V.S, Beberg A.L., Ensign D.L., Jayachdanran G., Khaliq S. Folding@home: Lessons From Eight Years of Volunteer Distributed Computing. ISPDC, 2009.

Roure D.D, Baker M.A, Jennings N.R, Shadbolt N.R, The Evolution of the Grid. Concurrency dan Computation: Practice dan Experience, 2003.

The Great Internet Mersenne Prime Search (GIMPS). 1996. [Online]. Available: http://www.mersenne.org/

The distributed.net website (DCTI). 1997. [Online]. Available: http://www.distributed.net/

Wang Yu, He Haiwu, Wang Zhijian. Towards a Formal Model of Volunteer Computing Systems. IDPDS, 2009.

Wu I-Chen, et al.,A Volunteer-Computing-Based Grid Environment for Connect6 Applications. CSE, 2009.

Yoshitomo Muratat, Tsutomu Inabatt , Hiroyuki Takizawat, Hiroaki Kobayashi. Implementation dan Evaluation of a Distributed dan Cooperative Load-Balancing Mechanism for Dependable Volunteer Computing. DSN, 2008.

Folding@home Wikipedia,

http://en.wikipedia.org/wiki/Folding@home [Online].

Tianhe-1 Wikipedia,

http://en.wikipedia.org/wiki/Tianhe-I [online].

ARM Cortex-A9 Wikipedia [online]

http://en.wikipedia.org/wiki/ARM_Cortex-A9_MPCore

Boincoid

http://boincoid.sourforge.net [online]