



SISTEM TERDISTRIBUSI

Ika Oktavia Suzanti, Husni

SISTEM TERDISTRIBUSI

Ika Oktavia Suzanti

Husni



SISTEM TERDISTRIBUSI

Penulis:
Ika Oktavia Suzanti
Husni

Editor:
Ika Oktavia Suzanti

Layouter :
Tim Kreatif PRCI

Cover:
Rusli

Cetakan Pertama : Oktober 2022

Hak Cipta 2022, pada Penulis. Diterbitkan pertama kali oleh:

Perkumpulan Rumah Cemerlang Indonesia
ANGGOTA IKAPI JAWA BARAT
Pondok Karisma Residence Jalan Raflesia VI D.151
Panglayungan, Cipedes Tasikmalaya – 085223186009

Website : www.rcipress.rcipublisher.org
E-mail : rumahcemerlangindonesia@gmail.com

Copyright © 2022 by Perkumpulan Rumah Cemerlang Indonesia
All Right Reserved

- Cet. I – : Perkumpulan Rumah Cemerlang Indonesia, 2022
; 14,8 x 21 cm
ISBN : 978-623-448-241-6

Hak cipta dilindungi undang-undang
Dilarang memperbanyak buku ini dalam bentuk dan dengan
cara apapun tanpa izin tertulis dari penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang
Hak Cipta Pasal 72

KATA PENGANTAR

Puja dan puji syukur penulis haturkan kepada Allah Subhanahu Wata'ala yang telah memberikan banyak nikmat, taufik, dan hidayah. Sehingga penulis dapat menyelesaikan Buku Ajar "Sistem Terdistribusi" dengan baik tanpa ada halangan yang berarti.

Buku ini telah penulis selesaikan berkat kerja sama dan bantuan dari beberapa pihak. Oleh karena itu penulis sampaikan banyak terima kasih kepada segenap pihak yang telah berkontribusi secara maksimal dalam penyelesaian buku ini. Buku ini merupakan aplikasi dari teori sistem distribusi dalam konteks sehari-hari.

Diluar itu penulis sebagai manusia biasa menyadari sepenuhnya bahwa masih banyak kekurangan dalam penulisan laporan ini, baik dari segi tata bahasa, susunan kalimat maupun isi. Oleh sebab itu dengan segala kerendahan hati, saya selaku penyusun menerima segala kritik dan saran yang membangun dari pembaca. Semoga buku ajar ini bisa menambah refrensi dan membawa manfaat bagi para pembaca.

Penulis

DAFTAR ISI

KATA PENGANTAR	3
DAFTAR ISI.....	4
BAB I KARAKTERISTIK SISTEM TERDISTRIBUSI.....	7
A. Pengertian.....	7
B. Trend Sistem Terdistribusi	12
C. Tantangan sistem terdistribusi	17
Latihan Soal.....	29
BAB II SISTEM MODEL.....	40
A. Layer Perangkat Lunak	41
B. Middleware.....	42
C. Sistem Arsitektur	44
D. Interaksi Model	47
E. Failure Model.....	52
F. Model Keamanan	56
Latihan Soal.....	60
BAB III INTERPROCESS COMMUNICATION.....	72
A. Komunikasi Synchronous and Asynchronous.....	73
B. Soket.....	75
C. Java API untuk UDP Datagram	80
D. Java API untuk TCP streams	83
E. Data Eksternal Representasi dan Marshalling	86
F. Group communication.....	99
Latihan Soal.....	104
BAB IV APPLICATION PROGRAMMING INTERFACE.....	121
A. Pengertian.....	121
B. Arsitektur API.....	125
C. Service Mesh	128
D. REST API	132
1. Membuat REST API Menggunakan PHP.....	132

2. Membangun REST API Menggunakan Framework (JAX-RS) Jersey.....	156
3. Membangun RESTful Web Service dan JAXB.....	178

DAFTAR TABEL

Gambar 1. Properti Sistem Terdistribusi.....	10
Gambar 2. Intranet.....	13
Gambar 3. <i>Ubiquitous computing</i>	16
Gambar 4. <i>Middleware Layers</i>	43
Gambar 5. Proses <i>Client - Server</i>	44
Gambar 6. <i>Invocation</i> Klien Server.....	57
Gambar 7. <i>Secure Channel</i>	60
Gambar 8. Komunikasi <i>Layer Middleware</i>	60
Gambar 9. <i>Socket Pairing</i>	75
Gambar 10. <i>SOAP Message Structure</i>	92
Gambar 11. Tampilan aplikasi <i>client</i>	97
Gambar 12. Komunikasi <i>Request-Reply</i>	99
Gambar 13. <i>System Application Programming Interface</i>	122
Gambar 14. <i>Service Mesh</i>	129

BAB I

KARAKTERISTIK SISTEM TERDISTRIBUSI

A. Pengertian

Jaringan komputer ada dimana-mana. Internet merupakan satu kesatuan, tanpa mengabaikan berapa banyak jumlah jaringan yang terhubung. Jaringan telepon seluler, perusahaan, pabrik, kampus, rumah, hingga semua perangkat pintar, baik secara terpisah maupun dalam jaringan yang sama menjadikan semua perangkat tersebut terhubung oleh internet.

Sistem terdistribusi adalah sebuah sistem (aplikasi) dimana komponennya terletak pada jaringan komputer. Semua komponen tersebut berkomunikasi dan mengkoordinasikan setiap tindakan dengan mengirimkan pesan. Komponen berinteraksi dengan lainnya untuk mencapai tujuan tertentu sesuai spesifikasi sistem. Sebuah sistem dapat disebut sebagai sistem terdistribusi apabila memiliki beberapa persyaratan sebagai berikut:

- *Concurrency* : Dalam jaringan komputer, aplikasi yang berjalan diatasnya harus dapat dieksekusi secara bersamaan. Sebagai contoh saya melakukan pekerjaan saya di komputer saya dan pada saat yang bersamaan orang lain melakukan pekerjaan di komputer mereka, dan saling berbagi sumber daya seperti halaman web atau file bila diperlukan. Kapasitas sistem untuk menangani sumber daya bersama dapat ditingkatkan dengan menambahkan lebih banyak sumber daya (misalnya komputer) ke jaringan. Pada buku ini akan dijelaskan cara-cara dimana kapasitas ekstra ini dapat

digunakan secara bergantian bahkan bersamaan. Koordinasi *running* program secara bersamaan yang berbagi sumber daya juga merupakan hal yang penting dan akan dibahas dalam beberapa bab dalam buku ini.

- *Tidak ada pengaturan waktu global*: Pada saat aplikasi membutuhkan untuk berkomunikasi dengan komponen lainnya maka dilakukan pertukaran pesan. Koordinasi yang erat sering kali bergantung pada waktu kapan pertukaran pesan itu dibutuhkan dan harus terjadi. Pada sisi lain, terdapat batasan waktu akurasi dengan komputer dalam jaringan agar dapat menyinkronkan terkait waktu yang tepat. Ini adalah konsekuensi langsung dari fakta bahwa satu-satunya cara untuk berkomunikasi adalah dengan mengirim pesan melalui sebuah jaringan.
- *Kegagalan independen* : Semua komputer sistem bisa gagal, dan itu tanggung jawab sistem desainer untuk merencanakan konsekuensi dari kemungkinan kegagalan. Sistem terdistribusi bisa gagal dalam keadaan yang berbeda. Kesalahan dalam jaringan mengakibatkan isolasi komputer yang terhubung dengannya, tetapi itu tidak berarti itu aplikasi berhenti. Kenyataannya, program-programnya mungkin tidak dapat mendeteksi apakah jaringan telah gagal atau menjadi sangat lambat. Demikian pula, kegagalan komputer, atau penghentian program yang tidak terduga di suatu tempat di sistem (*crash*), tidak segera diketahui oleh komponen lain yang digunakannya berkomunikasi. Setiap komponen sistem dapat gagal secara independen, sementara komponen lainnya tetap dapat berjalan.

Secara umum karakteristik sistem terdistribusi antar lain:

- Berbagi akses dan resource

Kemampuan untuk menggunakan setiap perangkat keras, perangkat lunak atau data dimana saja didalam sistem. *Resource manager* mengontrol akses, menyediakan skema penamaan dan kontrol konkurensi. *Resource* berbagi model (misalnya server klien atau berbasis objek) menggambarkan bagaimana resource disediakan, cara penggunaan dan cara berinteraksi antara penyedia layanan dan antar pengguna lainnya.

- Keterbukaan

Keterbukaan berhubungan dengan ekstensi dan perbaikan dari sistem terdistribusi. Detail *interface* dari komponen harus dipublikasikan. Komponen baru harus terintegrasi dengan yang komponen yang sudah ada. Perbedaan representasi data dari jenis *interface* pada prosesor yang berbeda (dari berbeda vendor) harus dapat terselesaikan.

- Konkurensi

Komponen dalam sistem terdistribusi dieksekusi dalam proses yang bersamaan (konkurensi proses). Komponen mengakses dan memperbarui resource bersama (misalnya variabel, database, perangkat driver). Integritas dari itu sistem mungkin dilanggar jika bersamaan pembaruan tidak dilakukan secara terkoordinasi sehingga terjadi lost updates atau analisis tidak konsisten.

- Skalabilitas

Adaptasi dari sistem terdistribusi untuk mengakomodasi banyak

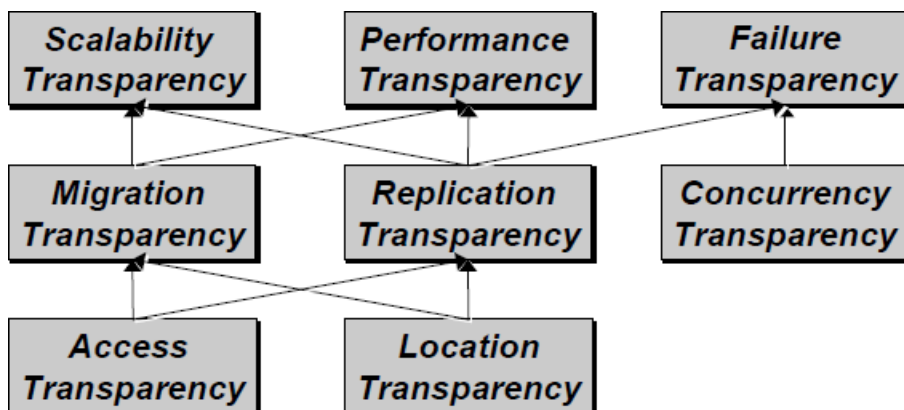
pengguna, menanggapi lebih cepat. Skalabilitas biasanya selesai oleh penambahan jumlah dan/atau prosesor lebih cepat. Komponen sebaiknya tidak berubah apabila skala dari sebuah sistem meningkat. Rancangan komponen harus dapat terukur.

- **Kesalahan Toleransi**

Sistem terdistribusi harus menjaga ketersediaan bahkan pada saat perangkat keras atau perangkat lunak atau keandalan jaringan berada pada level terendah. Kesalahan toleransi dapat disebabkan oleh recovery atau redundansi.

- **Transparansi**

Sistem terdistribusi sebaiknya dirasakan oleh pengguna dan programmer aplikasi sebagai sebuah kesatuan dibandingkan sebagai kumpulan dari komponen yang saling bekerja sama. Transparansi memiliki berbeda ukuran itu adalah diidentifikasi oleh ANSA. Beberapa properti yang harus dimiliki oleh sistem terdistribusi dan keterkaitan satu sama lain ditunjukkan pada gambar.



Gambar 1 Properti Sistem Terdistribusi

- Access Transparansi

Mengaktifkan lokal dan remote informasi objek agar dapat diakses menggunakan identik operasi. Contoh: File sistem operasi di NFS, Navigasi di Web, SQL Queries.

- Lokasi Transparansi

Mengaktifkan informasi objek agar dapat diakses tanpa harus diketahui letak lokasi. Contoh: File sistem operasi di NFS, Halaman di Web, tabel di database terdistribusi.

- Konkurensi Transparansi

Mengaktifkan beberapa proses agar dapat beroperasi bersamaan menggunakan bersama informasi objek tanpa gangguan antara masing-masing. Contoh: NFS, mesin kasir otomatis pada jaringan, DBMS.

- Replikasi Transparansi

Mengaktifkan banyak contoh dari informasi objek agar dapat digunakan untuk meningkat keandalan dan performa tanpa sepengetahuan replika oleh pengguna atau aplikasi program. Contoh: DBMS Terdistribusi, Mirroring halaman web.

- Kegagalan Transparansi

Memungkinkan penyembunyian kesalahan, Memungkinkan pengguna dan aplikasi untuk menyelesaikan tugas meskipun terjadi kegagalan dari komponen lainnya. Contoh: DBMS

- Migrasi Transparansi

Mengizinkan itu pergerakan dari informasi objek didalam sebuah sistem tanpa mempengaruhi itu operasi dari pengguna atau aplikasi program. Contoh: NFS, halaman web.

- Performa Transparansi

Mengizinkan sistem agar dapat dikonfigurasi ulang untuk memperbaiki performa sesuai dengan beban bervariasi.

- Penskalaan Transparansi

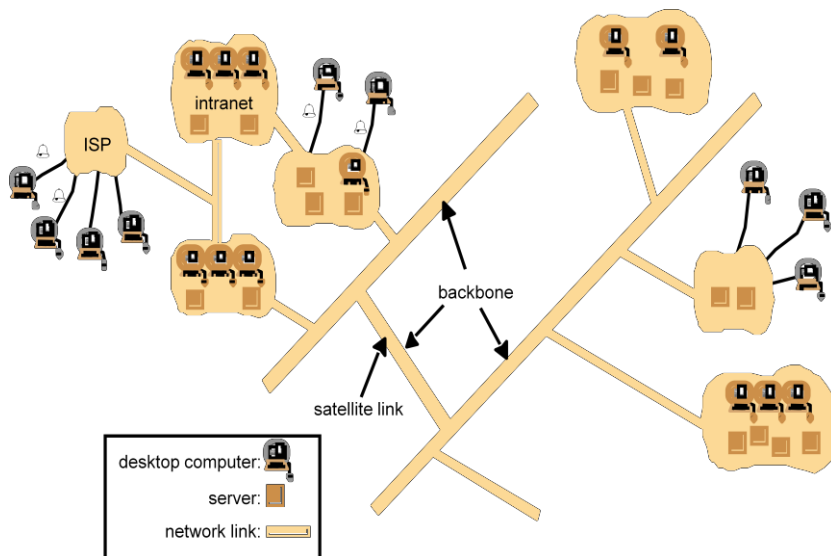
Mengizinkan itu sistem dan aplikasi memperluas dalam skala tanpa perubahan pada struktur sistem atau algoritma aplikasi.

Contoh: World Wide Web, basisdata terdistribusi.

B. Trend Sistem Terdistribusi

Sebuah sistem terdistribusi terdiri dari kumpulan komputer otonom, terhubung melalui jaringan dan distribusi middleware, yang memungkinkan komputer untuk mengoordinasikan kegiatan dan berbagi sumber daya sistem, jadi pengguna itu melihat itu sistem sebagai kesatuan fasilitas komputasi yang terintegrasi. Berikut perbedaan antara sistem terpusat dan sistem terdistribusi:

Sistem Terpusat	Sistem Terdistribusi
Satu komponen dengan bagian non-autonomous	Beberapa komponen autonomous
Komponen dibagi/digunakan bersama oleh pengguna setiap waktu	Komponen tidak dibagikan pada semua pengguna
Semua sumber daya dapat diakses	Sumber daya belum tentu dapat diakses
Perangkat lunak berjalan pada sebuah single proses	Perangkat lunak berjalan pada proses bersamaan di beberapa prosesor
Single point kontrol	Multiple point kontrol



Gambar 2. Intranet

Internet merupakan sistem terdistribusi yang sangat besar. Ini memungkinkan pengguna dimanapun berada, untuk membuat penggunaan layanan seperti World Wide Web, email dan transfer file. (Memang, Web adalah kadang-kadang salah disamakan dengan Internet) Kumpulan layanan bersifat terbuka bisa jadi diperpanjang dengan penambahan komputer server dan jenis layanan baru.

Gambar 2 menunjukkan kumpulan intranet – subjaringan yang dioperasikan oleh perusahaan dan organisasi lain dan biasanya dilindungi oleh firewall. Peran firewall adalah untuk melindungi intranet dengan mencegah pesan yang tidak sah dari meninggalkan atau memasuki. Sebuah firewall adalah penyaring pesan masuk dan keluar. Penyaringan mungkin dilakukan oleh sumber atau tujuan, atau firewall mungkin hanya mengizinkan pesan yang terkait dengan email dan akses web untuk masuk atau keluar dari intranet yang terlindungi. Penyedia Layanan Internet (ISP) adalah perusahaan yang menyediakan

link broadband dan jenis koneksi lain ke pengguna individu dan organisasi kecil, memungkinkan untuk mengakses layanan dimana saja di Internet serta menyediakan layanan lokal seperti email dan web hosting. Setiap intranet berhubungan bersama oleh *backbones*. Sebuah *backbones* adalah sebuah link jaringan dengan kapasitas transmisi tinggi, menggunakan koneksi satelit, kabel serat optik dan *bandwidth* sirkuit.

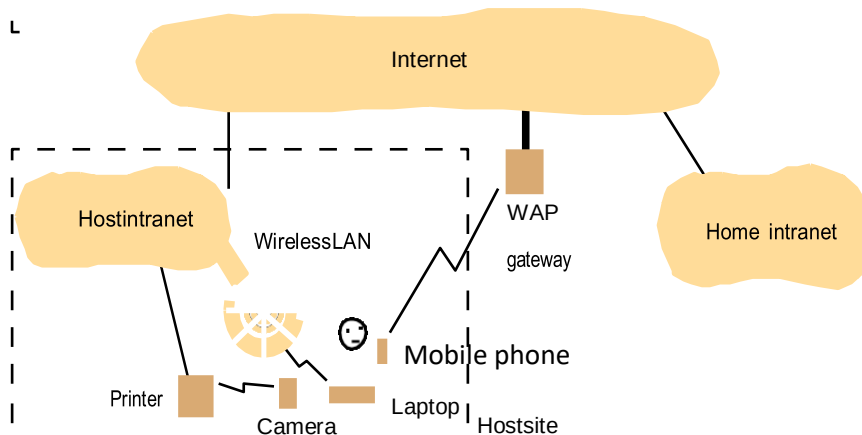
Contoh dari sistem terdistribusi dan aplikasi dari komputasi terdistribusi antara lain:

- Telekomunikasi jaringan,
- Jaringan telepon dan seluler jaringan,
- Jaringan komputer seperti Internet,
- Jaringan sensor nirkabel,
- Algoritma perutean,
- Aplikasi jaringan,
- Jaringan Worldwide web dan peer-to-peer,
- Komunitas game online dan virtual reality,
- Basisdata terdistribusi dan sistem manajemen Basisdata terdistribusi,
- Network File System
- Sistem pemrosesan informasi terdistribusi seperti sistem perbankan dan sistem reservasi penerbangan
- Real-time processing control seperti sistem kontrol penerbangan, sistem kontrol perindustrian
- Komputasi parallel seperti komputasi keilmuan meliputi komputasi cluster, komputasi grid ataupun proyek komputasi lainnya
- Rendering terdistribusi di komputer grafis.

Kemajuan teknologi dalam miniaturisasi perangkat dan jaringan nirkabel semakin mendorong ke integrasi perangkat komputasi kecil dan portabel menjadi sistem terdistribusi antara lain:

- Laptop komputer.
- Perangkat genggam, termasuk seluler telepon, telepon cerdas, perangkat GPS, pager, personal digital assistants (PDA), video kamera dan digital kamera.
- Perangkat wearable, seperti jam tangan cerdas dengan fungsionalitas serupa untuk PDA.
- Perangkat embedded pada peralatan rumah tangga seperti mesin cuci, sistem hi-fi, mobil dan lemari es.

Portabilitas dari banyak perangkat ini, bersama dengan kemampuannya untuk terhubung dengan nyaman ke jaringan di berbeda tempat, membuat seluler komputasi memungkinkan. Seluler komputasi adalah itu kinerja tugas komputasi saat pengguna sedang bepergian, atau mengunjungi tempat selain lingkungan biasanya. Dalam komputasi mobile, pengguna yang jauh dari intranet 'rumah' mereka (intranet pada pekerjaan, atau tempat tinggal mereka) masih diberikan akses ke sumber daya melalui perangkat yang mereka bawa. Mereka dapat terus mengakses Internet; mereka dapat terus mengakses sumber daya di intranet rumah mereka; dan ada peningkatan ketentuan bagi pengguna untuk memanfaatkan sumber daya seperti printer atau bahkan titik penjualan yang mudah dijangkau saat mereka bergerak. Yang terakhir juga dikenal sebagai location-aware atau context-aware computing. Mobilitas memperkenalkan sejumlah tantangan untuk sistem terdistribusi, termasuk kebutuhan untuk menangani konektivitas variabel dan pemisahan, dan kebutuhan untuk menjaga operasi tetap berjalan sebagai perangkat mobilitas.



Gambar 3. *Ubiquitous computing*

Ubiquitous computing adalah pemanfaatan banyak perangkat komputasi kecil dan murah yang umumnya di lingkungan fisik pengguna, termasuk rumah, kantor, dan bahkan pengaturan alami. Pengertian "*Ubiquitous*" dimaksudkan untuk menunjukkan bahwa perangkat komputasi kecil pada akhirnya akan menjadi begitu meresap dalam objek sehari-hari sehingga mereka hampir tidak diperhatikan. Artinya, perilaku komputasi mereka akan secara transparan dan erat terkait dengan fungsi fisiknya.

Kehadiran komputer di mana-mana hanya menjadi berguna ketika mereka dapat berkomunikasi dengan satu sama lain. Misalnya, mungkin nyaman bagi pengguna untuk mengontrol mesin cuci mereka atau sistem hiburan mereka dari telepon mereka atau perangkat 'remote control universal' di rumah. Sama, pencucian mesin dapat memberi tahu pengguna melalui pintar badge atau telepon ketika pencucian adalah selesai.

Komputasi di mana-mana dan seluler tumpang tindih, karena pengguna seluler pada prinsipnya dapat memperoleh manfaat dari komputer yang ada di mana-mana. Tapi mereka berbeda, secara umum. Komputasi di mana-mana bisa menguntungkan pengguna saat mereka tetap berada di satu lingkungan seperti rumah atau rumah sakit.

Demikian pula, komputasi mobile memiliki kelebihan meskipun hanya melibatkan komputer konvensional dan diskrit perangkat seperti sebagai laptop dan printer.

Berbagi resource:

- Merupakan motivasi utama dari komputasi terdistribusi
- Macam resource:
 - Hardware, misalnya printer, scanner, kamera
 - Data, misalnya file, basis data, webpage
 - Fungsi lebih spesifik, misalnya mesin pencari, file
- Service: Mengelola kumpulan resource terkait dan menyajikan fungsinya kepada pengguna dan aplikasi
- Server: Proses pada komputer jaringan yang menerima permintaan dari proses di komputer lain untuk melakukan layanan dan merespons dengan tepat
- Klien : Yang meminta proses

Sebuah interaksi lengkap antara klien dan server, dari saat klien mengirim permintaannya hingga saat menerima respons server

C. Tantangan sistem terdistribusi

Heterogenitas

Internet memungkinkan pengguna ke mengakses jasa dan menjalankan aplikasi pada kumpulan komputer dan jaringan. Heterogenitas (variasi dan perbedaan) berlaku untuk semua hal meliputi:

- jaringan;
- perangkat keras komputer;
- Sistem operasi;
- bahasa pemrograman;
- Implementasi oleh vendor berbeda

Meskipun Internet terdiri dari banyak jenis jaringan yang berbeda, perbedaannya tertutupi oleh fakta bahwa semua komputer yang terhubung dengannya menggunakan protokol Internet untuk berkomunikasi satu sama lain. Misalnya, komputer yang terhubung ke Ethernet memiliki implementasi dari: Protokol internet melalui Ethernet, sedangkan komputer pada jenis jaringan yang berbeda akan membutuhkan sebuah penerapan dari Internet protokol untuk jaringan tersebut.

Tipe data seperti bilangan bulat dapat direpresentasikan dengan cara yang berbeda pada berbagai jenis perangkat keras – misalnya, ada dua alternatif untuk pengurutan byte bilangan bulat. Perbedaan ini dalam representasi harus ditangani jika pesan akan dipertukarkan antara program yang berjalan perangkat keras yang berbeda. Meskipun sistem operasi semua komputer di Internet perlu: termasuk implementasi protokol Internet, mereka tidak harus semuanya menyediakan hal yang sama interface pemrograman aplikasi untuk protokol ini. Misalnya, panggilan untuk bertukar pesan di UNIX berbeda dari itu panggilan di windows.

Bahasa pemrograman yang berbeda menggunakan representasi yang berbeda untuk karakter dan struktur data seperti array dan record. Perbedaan ini harus diatasi jika program ditulis dalam bahasa yang berbeda bahasa adalah ke menjadi sanggup ke menyampaikan dengan satu lain. Program ditulis oleh pengembang berbeda tidak bisa berkomunikasi dengan satu lain kecuali mereka menggunakan standar umum, untuk contoh, jaringan komunikasi dan representasi item data primitif dan struktur data di pesan. Agar ini dapat terjadi, standar membutuhkan persetujuan dan diadopsi – seperti pada internet protokol.

Middleware

Istilah middleware berlaku untuk lapisan perangkat lunak yang menyediakan pemrograman abstraksi serta menutupi heterogenitas jaringan, perangkat keras, operasi yang mendasarinya sistem dan bahasa pemrograman. Common Object Request Broker (CORBA), adalah sebuah contoh. Beberapa middleware, seperti Java Remote Method Invocation (RMI), hanya mendukung sebuah bahasa pemrograman tunggal. Kebanyakan middleware diimplementasikan melalui protokol Internet, dimana masing-masing menutupi perbedaan yang mendasarinya jaringan, tetapi semua middleware harus bisa mengatasi perbedaan pada sistem operasi dan perangkat keras.

Heterogenitas dan seluler kode

Itu ketentuan seluler kode adalah digunakan untuk merujuk ke kode program yang dapat ditransfer dari satu komputer ke komputer lain dan dijalankan di tempat tujuan – Java applet adalah sebuah contoh. Kode yang cocok untuk dijalankan di satu komputer belum tentu cocok untuk dijalankan di tempat lainnya karena program yang dapat dieksekusi biasanya spesifik untuk set instruksi dan untuk host sistem operasi.

Pendekatan mesin virtual menyediakan cara untuk membuat kode dapat dieksekusi pada berbagai host komputer: compiler untuk sebuah bahasa tertentu menghasilkan kode untuk sebuah virtual mesin alih-alih dari sebuah tertentu perangkat keras. Sebagai contoh, Java compiler menghasilkan kode untuk sebuah Java virtual mesin, yang mengeksekusinya oleh interpretation.

Mesin virtual Java perlu diimplementasikan sekali untuk setiap jenis komputer untuk mengaktifkan Java program agar dapat di-*running*. Saat ini, itu paling umumnya digunakan membentuk dari

seluler kode adalah itu penyertaan Javascript program di beberapa web halaman lalu ditampilkan pada browser klien.

Keterbukaan (Openness)

Keterbukaan suatu sistem komputer merupakan karakteristik yang menentukan apakah sistem tersebut dapat menjadi extend dan diimplementasikan kembali dengan berbagai cara. Keterbukaan dari sistem terdistribusi ditentukan terutama oleh sejauh mana layanan berbagi sumber daya baru dapat ditambahkan dan tersedia untuk digunakan oleh berbagai variasi dari program klien.

Keterbukaan tidak dapat dicapai kecuali spesifikasi dan dokumentasi perangkat lunak utama interface komponen sistem tersedia untuk pengembang perangkat lunak. Dengan kata lain, kunci interface dipublikasikan. Proses ini mirip dengan standarisasi interface, tetapi sering dilewati oleh standardisasi prosedur yang biasanya rumit dan bergerak lambat. Namun, publikasi interface hanya titik awal untuk menambah dan memperluas layanan dalam sistem terdistribusi. Tantangan untuk desainer adalah untuk mengatasi kompleksitas sistem terdistribusi yang terdiri dari banyak komponen direkayasa oleh orang yang berbeda. Perancang protokol Internet memperkenalkan serangkaian: dokumen yang disebut 'Requests for Comments', atau RFC, yang masing-masing dikenal dengan nomor. Itu spesifikasi protokol komunikasi Internet diterbitkan dalam seri ini di awal 1980-an, diikuti dengan spesifikasi untuk aplikasi yang menjalankannya, seperti transfer file, email dan telnet oleh itu pertengahan 1980-an.

Sistem yang dirancang untuk mendukung berbagi sumber daya dengan cara ini disebut terdistribusi terbuka sistem untuk menekankan fakta bahwa mereka dapat diperluas. Mereka dapat diperpanjang di perangkat keras tingkat dengan penambahan komputer ke jaringan dan

di tingkat perangkat lunak dengan pengenalan layanan baru dan implementasi ulang layanan sebelumnya, memungkinkan program aplikasi untuk berbagi resource.

Open sistem adalah karakteristik berdasarkan fakta bahwa kunci interface dipublikasikan. Sistem terdistribusi terbuka didasarkan pada penyediaan mekanisme komunikasi yang seragam dan diterbitkan interface untuk mengakses ke sumber daya bersama. Sistem terdistribusi terbuka dapat dibangun dari heterogen perangkat keras dan perangkat lunak, mungkin dari berbeda vendor. Tetapi itu kesesuaian dari masing-masing komponen itu harus distandarisasi, hati-hati diuji dan diverifikasi jika itu sistem dapat bekerja dengan benar.

Keamanan

Banyak sumber daya informasi yang tersedia dan dipelihara dalam sistem terdistribusi memiliki nilai intrinsik yang tinggi bagi penggunanya. Oleh karena itu, keamanan mereka sangat penting. Keamanan untuk sumber informasi memiliki tiga komponen: kerahasiaan (perlindungan terhadap pengungkapan kepada individu tidak sah), integritas (perlindungan melawan perubahan atau data yang korup), dan ketersediaan (perlindungan melawan gangguan dengan sarana untuk mengakses sumber daya). Di sebuah sistem terdistribusi, klien mengirim permintaan untuk mengakses data yang dikelola oleh server, yang melibatkan pengiriman pesan ke sebuah jaringan. Contoh:

1. Seorang dokter mungkin meminta mengakses ke data pasien RSUD atau kirim tambahan pada data tersebut.
2. Di elektronik perdagangan dan perbankan, pengguna kirim nomor kartu kredit mereka melalui kartu Internet.

Pada kedua contoh, tantangannya adalah mengirim informasi sensitif dalam pesan melalui jaringan cara yang aman. Tapi keamanan bukan hanya soal menyembunyikan isi pesan –melainkan juga melibatkan mengetahui dengan pasti identitas pengguna atau agen lain atas nama siapa pesan itu terkirim. Namun, dua tantangan keamanan belum pernah sepenuhnya tertangani:

- Serangan Denial of service: masalah keamanan lainnya adalah sebuah pengguna mungkin akan mengganggu sebuah layanan untuk beberapa alasan. Ini dapat dicapai dengan membombardir layanan dengan sejumlah besar permintaan sia-sia sehingga pengguna yang serius tidak dapat menggunakannya. Ini disebut serangan denial of service. Ada beberapa serangan denial of service pada layanan web terkenal. Saat ini serangan tersebut dilawan dengan mencoba untuk menangkap dan menghukum pelaku setelah kejadian, tapi itu bukan merupakan solusi umum untuk masalah tersebut.
- Keamanan kode seluler: Kode seluler harus ditangani dengan hati-hati. Pertimbangkan seseorang yang menerima program yang dapat dieksekusi sebagai lampiran surat elektronik: kemungkinan efek menjalankan program tidak dapat diprediksi; misalnya, mungkin tampak menampilkan gambar yang menarik tetapi dalam realitas dia mungkin mengakses lokal sumber daya, atau mungkin menjadi bagian dari sebuah penolakan dari serangan layanan.

Skalabilitas

Sistem terdistribusi beroperasi secara efektif dan efisien pada berbagai skala, mulai dari sebuah intranet kecil ke Internet. Sebuah sistem digambarkan sebagai scalable akan tetap efektif ketika ada peningkatan yang signifikan dalam jumlah sumber daya dan jumlah

pengguna. Jumlah komputer dan server di Internet telah meningkat secara dramatis. Gambar 2 menunjukkan peningkatan nomor komputer dan jaringan server selama itu 12 tahun sejarah Web ke atas ke 2005 [zakon.org]. Sangat menarik untuk dicatat pertumbuhan yang signifikan di komputer dan server web di periode ini, tetapi juga bahwa persentase relatif mendatar – tren yang dijelaskan oleh pertumbuhan komputasi pribadi tetap dan seluler. Satu server web juga dapat di-host semakin banyak komputer.

Desain sistem terdistribusi scalable menghadirkan tantangan berikut:

- Mengontrol biaya sumber daya fisik: Seiring dengan meningkatnya permintaan akan sumber daya, hal itu harus mungkin untuk memperluas sistem, dengan biaya yang wajar untuk memenuhinya. Misalnya, frekuensi dengan file mana yang diakses di intranet kemungkinan akan bertambah seiring jumlah pengguna dan komputer meningkat. Harus dimungkinkan untuk menambahkan komputer server untuk menghindari kemacetan kinerja yang akan muncul jika satu file server harus menangani semua permintaan akses file. Secara umum, untuk sistem dengan n pengguna menjadi terukur, itu kuantitas dari sumber daya fisik diperlukan untuk mendukung mereka Sebaiknya menjadi paling banyak $O(n)$ – yaitu sebanding dengan n . Misalnya, jika satu file server dapat mendukung 20 pengguna, maka dua server harus bisa mendukung 40 pengguna.
- Mengontrol kehilangan kinerja: Pertimbangkan pengelolaan satu set data yang ukurannya sebanding dengan jumlah pengguna atau sumber daya dalam sistem – misalnya, tabel dengan korespondensi antara nama domain komputer dan

alamat Internet mereka yang dipegang oleh Domain Nama Sistem, yang digunakan terutama untuk mencari nama DNS seperti `www.amazon.com`. Algoritma yang menggunakan struktur hierarki skala lebih baik daripada yang menggunakan struktur linier. Tetapi bahkan dengan struktur hierarkis, peningkatan ukuran akan hasil di beberapa kehilangan kinerja: waktu yang dibutuhkan untuk mengakses data yang terstruktur secara hierarkis adalah $O(\log n)$, di mana n adalah ukuran dari itu mengatur dari data. Untuk sebuah sistem ke menjadi terukur, maksimum pertunjukan kehilangan Sebaiknya menjadi Tidak lebih buruk dibandingkan ini.

- Mencegah kehabisan sumber daya perangkat lunak : Contoh kurangnya skalabilitas ditunjukkan oleh nomor yang digunakan sebagai alamat Internet (IP) (alamat komputer di Internet). Pada akhir 1970-an, itu diputuskan untuk menggunakan 32bit untuk tujuan ini, tetapi seperti yang akan dijelaskan di Bab 3, suplai alamat Internet yang tersedia hampir habis. Untuk alasan ini, versi baru dari protokol dengan Alamat Internet 128-bit sedang diadopsi, dan ini akan membutuhkan modifikasi pada banyak komponen perangkat lunak.
- Menghindari kemacetan: Umumnya, algoritma sebaiknya menjadi terdesentralisasi untuk menghindari hambatan kinerja. Kami mengilustrasikan hal ini dengan mengacu pada pendahulu dari Domain Name System, di mana tabel nama disimpan dalam satu file master yang bisa diunduh ke setiap komputer itu diperlukan dia. Itu dulu baik-baik saja ketika hanya ada beberapa ratus komputer di Internet, tetapi seiring waktu menjadi serius kinerja dan administratif kemacetan.

Kegagalan penanganan

Sistem komputer terkadang gagal. Ketika kesalahan terjadi pada perangkat keras atau perangkat lunak, program dapat: menghasilkan hasil yang salah atau mungkin berhenti sebelum mereka menyelesaikan perhitungan yang dimaksudkan. Kegagalan dalam sistem terdistribusi bersifat parsial – yaitu, beberapa komponen gagal sementara yang lain berlanjut ke fungsi. Karena itu itu penanganan dari kegagalan terutama sulit.

- Mendeteksi kegagalan: Beberapa kegagalan dapat dideteksi. Misalnya, checksum dapat digunakan untuk mendeteksi rusaknya data pada sebuah pesan atau file. Terdapat kesulitan atau bahkan mustahil ke mendeteksi beberapa kegagalan lainnya, seperti crash pada server jarak jauh di Internet. Tantangannya adalah mengelola kegagalan yang tidak dapat dideteksi tetapi dapat dicurigai.
- Kegagalan masking: Beberapa kegagalan itu pernah terdeteksi dapat tersembunyi atau dibuat lebih sedikit berat. Dua contoh dari kegagalan tersembunyi:
 1. Pesan dapat ditransmisikan ulang apabila mereka gagal sampai di tujuan.
 2. Mengajukan data bisa menjadi tertulis pada sebuah disk berpasangan sehingga jika satu rusak, maka lainnya mungkin tetap benar.
- Kegagalan toleransi: Sebagian besar layanan di Internet memang menunjukkan kegagalan – itu tidak akan terjadi praktis bagi mereka untuk mencoba mendeteksi dan menyembunyikan semua kegagalan yang mungkin terjadi dalam skala besar jaringan dengan begitu banyak komponen.

Klien mereka dapat dirancang untuk mentolerir kegagalan, yang umumnya melibatkan pengguna yang menoleransi mereka juga. Misalnya, ketika browser web tidak dapat hubungi server web, itu tidak membuat pengguna menunggu selamanya saat terus mencoba – ini memberi tahu pengguna tentang masalah, membiarkan mereka bebas untuk mencoba lagi nanti.

- Pemulihan dari kegagalan : Pemulihan melibatkan desain perangkat lunak sehingga keadaan permanen data dapat dipulihkan atau “rollback” setelah server crash. Secara umum, perhitungan dilakukan oleh beberapa program tidak akan lengkap ketika kesalahan terjadi, dan data permanen itu akan memperbarui (berkas dan lainnya bahan disimpan di penyimpanan permanen) mungkin tidak konsisten.
- Redundansi: Layanan dapat dibuat untuk mentolerir kegagalan dengan menggunakan komponen yang berlebihan. Berikut sebagai contoh:
 1. Sebaiknya selalu ada pada paling sedikit dua rute berbeda di antara setiap dua router di Internet.
 2. Pada Domain Nama Sistem, setiap tabel nama direplikasi pada setidaknya dua server berbeda.
 3. Basis data dapat direplikasi di beberapa server untuk memastikan bahwa data tetap dapat diakses setelah kegagalan server tunggal mana pun; server dapat dirancang untuk mendeteksi kesalahan pada rekan-rekan mereka; Ketika sebuah kesalahan terdeteksi satu peer, klien dialihkan ke sisa server.

Konkurensi

Baik layanan dan aplikasi menyediakan sumber daya yang dapat digunakan bersama oleh klien secara terdistribusi sistem. Di sana oleh karena itu kemungkinan bahwa beberapa klien akan mencoba ke mengakses berbagi sumber daya secara bersamaan. Misalnya, struktur data yang mencatat tawaran untuk lelang mungkin: sangat sering diakses ketika mendekati batas waktu. Proses yang mengelola sumber daya bersama dapat mengambil satu permintaan klien pada satu waktu. Tapi pendekatan itu membatasi throughput. Oleh karena itu layanan dan aplikasi umumnya memungkinkan beberapa permintaan klien untuk diproses bersamaan. Untuk membuat ini lebih konkret, anggaplah setiap sumber daya dienkapsulasi sebagai objek dan bahwa pemanggilan dieksekusi dalam threads bersamaan. Dalam hal ini mungkin saja beberapa threads dapat dieksekusi secara bersamaan dalam suatu objek, dalam hal ini operasinya mungkin akan konflik satu lain dan menghasilkan hasil tidak konsisten.

Transparansi

Transparansi didefinisikan sebagai penyembunyian dari pengguna dan pemrogram aplikasi dari pemisahan komponen-komponen dalam suatu sistem terdistribusi, sehingga sistem tersebut dirasakan secara keseluruhan bukan sebagai kumpulan komponen independen. Implikasi dari transparansi merupakan pengaruh besar pada rancangan dari sistem perangkat lunak.

- Mengakses transparansi memungkinkan lokal dan sumber daya terbatas diakses menggunakan operasi identik.
- Lokasi transparansi memungkinkan sumber daya diakses tanpa sepengetahuan dari fisik atau jaringan lokasi (contoh, untuk bangunan atau alamat IP). Konkurensi transparansi

memungkinkan beberapa proses agar beroperasi secara konkuren bersama sumber daya tanpa gangguan di antara mereka.

- Transparansi replikasi memungkinkan banyak sumber daya digunakan untuk meningkatkan keandalan dan kinerja tanpa pengetahuan dari replika oleh pengguna atau aplikasi programmer.
- Transparansi kegagalan memungkinkan penyembunyian kesalahan, memungkinkan pengguna dan program aplikasi ke menyelesaikan milik mereka tugas meskipun gagal perangkat keras atau komponen perangkat lunak.
- Mobilitas transparansi memungkinkan itu pergerakan dari sumber daya dan klien di dalam sebuah sistem tanpa mempengaruhi itu operasi dari pengguna atau program.
- Performa transparansi memungkinkan sistem dikonfigurasi ulang untuk meningkatkan performa seiring variasi beban.
- Penskalaan transparansi memungkinkan sistem dan aplikasi dikembangkan tanpa perubahan pada struktur sistem atau algoritma aplikasi.

Quality of Service

Setelah pengguna diberikan fungsionalitas yang mereka butuhkan dari suatu layanan, seperti file layanan dalam sistem terdistribusi, kita bisa bertanya tentang kualitas layanan yang diberikan. Itu sifat nonfungsional utama dari sistem yang mempengaruhi kualitas layanan yang dialami oleh klien dan pengguna adalah keandalan, keamanan dan kinerja. Kemampuan beradaptasi perubahan sistem konfigurasi dan ketersediaan resource sebagai sebuah aspek jauh lebih penting dari melayani kualitas.

Beberapa aplikasi, termasuk aplikasi multimedia, menangani data kritis waktu – aliran data yang diperlukan untuk diproses atau ditransfer dari satu proses ke proses lainnya pada tingkat yang tetap. Untuk contoh, layanan film mungkin terdiri dari program klien yang mengambil film dari video server dan menampilkannya di layar pengguna. Untuk hasil yang memuaskan, frame berturut-turut dari video membutuhkan ke ditampilkan ke pengguna di dalam beberapa waktu yang ditentukan batas.

Di faktanya, singkatan QoS telah secara efektif dikomandoi ke merujuk dengan kemampuan sistem untuk memenuhi tenggat waktu tersebut. Pencapaiannya tergantung pada ketersediaan yang diperlukan komputasi dan sumber daya jaringan pada waktu yang tepat. Ini menyiratkan persyaratan untuk sistem untuk memberikan jaminan komputasi dan sumber daya komunikasi yang cukup untuk memungkinkan aplikasi untuk menyelesaikan setiap tugas tepat waktu (misalnya, tugas menampilkan frame) video).

Latihan Soal

1. Berikan lima jenis sumber daya perangkat keras dan lima jenis sumber daya data atau perangkat lunak yang berguna untuk dibagikan. Berikan contoh pembagian mereka seperti yang terjadi dalam sistem terdistribusi.

Jawaban

- Perangkat keras:
CPU: server komputasi (mengeksekusi aplikasi intensif prosesor untuk klien), server objek jarak jauh (mengeksekusi metode atas nama klien), program worm (berbagi kapasitas cpu mesin