

Nama : Ari Satrio S.
NPM : 183510488
Kelas : Teknik Informatika 6E

Sistem Terdistribusi

Karakteristik Sistem Terdistribusi

a. Middleware

Middleware merupakan komponen perantara yang memungkinkan client dan server (lapisan aplikasi dan sistem operasi) saling terhubung dan berkomunikasi satu sama lain. Tujuan utama layanan middleware adalah untuk membantu memecahkan interkoneksi beberapa aplikasi dan masalah interoperabilitas. Middleware sangat dibutuhkan untuk bermigrasi dari aplikasi mainframe ke aplikasi client/server dan juga untuk menyediakan komunikasi antar platform yang berbeda.

Contoh middleware antara lain: Java's: Remote Procedure Call, Object Management Group's: Common Object Request Broker Architecture (CORBA), Microsoft's COM/DCOM (Component Object Model/ .NET Remoting), ActiveX controls (in-process COM components).

b. Manfaat Sistem Terdistribusi

- Data Sharing: Mengijinkan pengguna untuk bisa mengakses data yang sama.
- Device Sharing: Mengijinkan pengguna untuk bisa mengakses perangkat keras yang sama.
- Communication: Memungkinkan pengguna bisa melakukan komunikasi jauh lebih mudah.
- Multiuser Computing: Menerapkan banyak user yang dapat login pada saat yang bersamaan untuk mengakses sistem.
- Flexibility: 1) Membagi beban kerja pada perangkat yang tersedia dengan cara yang efektif. 2) Dapat menambah komponen secara individu tanpa harus menduplikasi sistem. 3) Fasilitas local dapat disesuaikan dengan kebutuhan local. 4) Memungkinkan pertumbuhan sistem secara terus menerus. 5) Susunan sistem bisa disesuaikan dengan pola organisasi perusahaan. 6) Memungkinkan beberapa bagian/local mengadakan percobaan konsep baru dan fasilitas baru untuk mengurangi resiko kegagalan sistem secara keseluruhan.

Dari manfaat tersebut dapat diperoleh kesimpulan bahwa perbedaan sistem tersebar dengan sistem terpusat, sehingga perlu adanya sistem tersebar yaitu:

- Resource Sharing.

- Computation Speedup
- Reliability
- Communication.

c. Kerugian Sistem Terdistribusi

- Complexity
Secara umum secara rancangan, implementasi dan maintenance (*error handdling*) jadi lebih kompleks daripada arsitektur terpusat.
- Security
Asumsi terhadap sistem yang terhubung jaringan artinya telah masuk ke jaringan publik ini artinya setiap orang diperbolehkan untuk bisa mengakses data pada jaringan publik. Asumsi ini bisa menjadi lubang keamanan terhadap pengaksesan data secara tidak sah.
- Manageability
Membutuhkan banyak usaha untuk mengorganisasi sistem.
- Unpredictability
Sulit diprediksi terhadap ketidakstabilan sistem tergantung dari sistem organisasi dan banyaknya akses terhadap jaringan.

d. Karakteristik Sistem Terpusat dan Sistem Terdistribusi

Sistem Terpusat:

- Komputer otonom yang secara eksplisit (diberikan alamat ip)
- Berbagai resource setiap saat untuk seluruh user
- Aplikasi berjalan diatas prosessor tunggal
- Sumber daya terlihat/nampak
- Pengendalian tunggal
- *Single point of failure*

Sistem Terdistribusi:

- Gabungan komputer bersifat otonom
- Sumber daya tidak dapat di akses secara langsung
- Aplikasi berjalan secara bersamaan pada prosessor yang berbeda
- *Concurrency of components.*
- *No global clock.*
- *Independent failures of components.*
- *Components are not shared by all users*
- *Multiple Points of control*

e. Tantangan

1) *Making resource accessible*

Hal utama dalam sistem tersebar adalah mudahnya pengguna melakukan akses terhadap sumber daya. Sumber daya dapat berbentuk *hardware* maupun *software*. Menghubungkan beberapa user dengan sumber daya memudahkan untuk proses penggabungan dan pertukaran informasi. Sebagai contoh yaitu printer, fasilitas penyimpanan, data, files, halaman web dan lain-lain.

2) *Distribution Transparency*

Tujuan penting dari sistem terdistribusi adalah menyembunyikan bahwa fakta pemrosesan suatu *job* dilakukan secara tersebar melalui beberapa perangkat fisik yang berbeda (*Transparency*). *Transparency* membuat beberapa aspek distribusi tidak nampak oleh pengguna. Aspek-aspek tersebut meliputi:

- *Transparency of access*; Pemetaan drive di samba, NFS, remotepriinter.
- *Transparency of location*; Transaksi perbankan ATM, Web Pages dll.
- *Transparency of relocation*; www, and Distributed database.
- *Transparency of replication*; Distributed DBMS, Mirroring Web Pages.
- *Transparency of concurrency*; NFS, Automatic teller machine network, Database management system.
- *Transparency of failure*; Database management system.
- *Transparency of Performance*;
- *Transparency of Migration*; NFS, Web pages.

3) *Openness*

Memastikan sistem dapat diperluas dan mudah dalam pemeliharaan. Kunci aspek dari *openness* adalah:

- Antarmuka standar dan protocol (seperti protokol komunikasi internet).
- Mendukung keberagaman (heterogeneity).

4) *Scalability*

Bertambahnya kebutuhan terhadap sistem, memungkinkan untuk memperbesar dan memperkecil infrastruktur dari sistem tersebar itu sendiri.

Scalability Problems:

- Centralized services: satu service untuk semua requests.
- Centralized data: satu data (data point) untuk semua services.
- Centralized algorithms: satu komputasi untuk semua requests.

Teknik menyelesaikan permasalahan skala:

- Hiding Communication Latencies (mengoptimalkan transmisi)
- Distribution (contoh: DNS)
- Replication (copy dan cache, masalah konsistensi)

5) *Concurrency*

Banyak client yang mengakses banyak data dalam waktu yang bersamaan, sedangkan data harus tetap konsisten. Proses dilakukan secara simultan dan membagi sumber daya sehingga bisa digunakan secara bersama untuk menghindari masalah deadlock.

6) *Fault Tolerance*

Kemampuan untuk tetap melakukan komputasi ketika terjadi kesalahan. Contohnya yaitu deteksi/mask/toleransi kesalahan, recovery terhadap kesalahan dan *redundancy*.

f. Jenis Sistem Terdistribusi

1) Distributed Computing System

- *Cluster Computing Systems*
- *Grid Computing Systems*

2) Distributer Information System

- *Transaction Processing Systems*
- *Enterprise Application Integration (Exchange info via RPC or RMI)*

3) Distributed Pervasive Systems (usually small, battery powered systems, mobile & wireless)

- Home Systems (e.g. Smart phones, PDAs)
- Electronic Health care systems (Heart monitors, BAN: Body Area Networks)
- Sensor Networks (distributed Databases connected wirelessly)

g. Grid Computing System

Sebagai bandingan, arsitektur lain yang telah lebih dulu ada (parallel computer maupun Cluster computing) umumnya bekerja di atas lingkungan yang homogenous dan terdedikasi. Untuk mengaktifkan Grid Computing, diperlukan perangkat lunak Middleware khusus.

Kondisi saat ini:

- Komputer-komputer penyedia sumber daya bersifat heterogenous karena terdiri dari berbagai jenis perangkat keras, sistem operasi maupun aplikasi yang terpasang.
- Komputer-komputer terhubung ke jaringan yang luas dengan kapasitas bandwidth yang beragam.
- Komputer maupun jaringan tidak terdedikasi, bisa hidup / mati tidak sewaktu-waktu tanpa jadwal yang jelas.