



LAPORAN
Sistem Terdistribusi
“Komunikasi Dalam Sistem Terdistribusi”
SEMESTER 5

DISUSUN OLEH
Muhammad Bintang
TRPL3C
2111082027

JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK
POLITEKNIK NEGERI PADANG
2023

Ringkasan Materi

Komunikasi dalam sistem terdistribusi adalah proses pertukaran informasi atau data antar-komputer atau komponen dalam jaringan terdistribusi.

Ini memungkinkan berbagai bagian sistem terdistribusi untuk saling berinteraksi dan bekerja bersama untuk mencapai tujuan bersama.

Komunikasi ini terjadi melalui jaringan komputer dan dapat melibatkan pengiriman pesan, data, atau sumber daya lainnya antar-node dalam sistem terdistribusi.

Tujuan utama adalah memungkinkan kolaborasi, koordinasi, dan berbagi sumber daya di seluruh sistem terdistribusi untuk meningkatkan kinerja dan efisiensi keseluruhan.

Apa Pentingnya Sistem Terdistribusi?

Koordinasi dan Kolaborasi: Sistem terdistribusi terdiri dari banyak komponen yang bekerja bersama untuk mencapai tujuan tertentu. Komunikasi memungkinkan komponen-komponen ini untuk berkoordinasi dan berkolaborasi, sehingga tugas-tugas dapat diselesaikan dengan efisien.

Berbagi Sumber Daya: Dalam sistem terdistribusi, sumber daya seperti data, perangkat keras, atau layanan mungkin tersebar di seluruh jaringan. Komunikasi memungkinkan akses yang aman dan efisien ke sumber daya ini, memaksimalkan penggunaan mereka.

Meningkatkan Kinerja: Komunikasi yang baik dapat meningkatkan kinerja sistem terdistribusi dengan memungkinkan transfer data yang cepat dan efisien antar komponen. Ini penting terutama dalam aplikasi yang membutuhkan pemrosesan data real-time atau transaksi tingkat tinggi.

Skalabilitas: Komunikasi yang baik juga memungkinkan sistem untuk ditingkatkan dengan mudah. Ketika jumlah pengguna atau beban kerja meningkat, sistem terdistribusi dapat menyesuaikan diri dengan cepat untuk mengatasi peningkatan tersebut.

Redundansi dan Keandalan: Komunikasi yang baik dapat menyediakan mekanisme redundansi untuk melindungi sistem terdistribusi dari kegagalan. Ketika salah satu komponen gagal, komunikasi yang andal memastikan bahwa tugas-tugas dapat beralih ke komponen lain tanpa gangguan yang signifikan.

Fleksibilitas dan Adaptabilitas: Lingkungan bisnis dan teknologi selalu berubah. Komunikasi yang efektif memungkinkan sistem terdistribusi untuk beradaptasi dengan perubahan ini dengan mudah, seperti penambahan atau penggantian komponen.

Keselarasan dan Konsistensi Data: Penting untuk memastikan bahwa data yang digunakan oleh berbagai komponen dalam sistem terdistribusi tetap konsisten. Komunikasi yang baik memungkinkan untuk sinkronisasi data dan menjaga keselarasan di seluruh sistem.

Keamanan: Keamanan adalah faktor kunci dalam sistem terdistribusi. Komunikasi yang baik melibatkan pengamanan data dan perlindungan terhadap ancaman keamanan

Apa Saja Protokol Komunikasi?

Standardisasi: Protokol komunikasi mendefinisikan tata cara yang konsisten untuk berkomunikasi di seluruh sistem, memastikan bahwa semua komponen dapat saling memahami.

Interoperabilitas: Membantu komponen yang berbeda, termasuk perangkat keras dan perangkat lunak dari vendor yang berbeda, untuk berkomunikasi tanpa masalah.

Kontrol dan Pengamanan: Protokol dapat mengatur cara data dikirim, diterima, dan dilindungi. Ini termasuk pengaturan keamanan, autentikasi, dan enkripsi data.

Keandalan: Protokol dapat memastikan bahwa pesan atau data yang dikirim tiba dengan benar dan dalam urutan yang benar, serta mengatasi masalah jika terjadi kegagalan dalam pengiriman.

Efisiensi: Protokol yang dirancang dengan baik dapat meminimalkan overhead komunikasi, seperti latensi dan penggunaan bandwidth, yang penting dalam sistem terdistribusi yang skalabel.

Apa Middleware? Apa saja contohnya?

Middleware adalah perangkat lunak yang berfungsi sebagai perantara atau jembatan antara berbagai komponen perangkat lunak atau aplikasi yang berbeda dalam sebuah sistem terdistribusi. Middleware memungkinkan komunikasi, koordinasi, dan interaksi antara komponen-komponen tersebut tanpa harus mengetahui detail implementasi masing-masing. Ini adalah lapisan perangkat lunak yang menyediakan layanan abstrak dan umum untuk mendukung integrasi sistem terdistribusi.

Message-Oriented Middleware (MOM): Middleware berbasis pesan digunakan untuk mengirim pesan antar-komponen dalam sistem terdistribusi. Contoh MOM termasuk Apache Kafka, RabbitMQ, dan Apache ActiveMQ.

Remote Procedure Call (RPC) Middleware: RPC middleware memungkinkan pemanggilan fungsi atau prosedur di mesin yang berbeda melalui jaringan seperti jika mereka adalah pemanggilan lokal. Contoh termasuk gRPC, CORBA, dan Java RMI.

Object Request Broker (ORB): Middleware ORB digunakan untuk komunikasi antar-objek dalam sistem yang menggunakan arsitektur berorientasi objek. Contoh middleware ORB termasuk TAO (The ACE ORB) dan ORBacus.

Database Middleware: Middleware database memfasilitasi akses dan komunikasi dengan basis data dalam sistem terdistribusi. Contoh middleware database termasuk JDBC (Java Database Connectivity) dan ODBC (Open Database Connectivity).

Web Middleware: Ini adalah jenis middleware yang mendukung pengembangan dan pengiriman aplikasi web. Contoh middleware web termasuk Apache Tomcat (untuk servlet Java) dan Microsoft ASP.NET (untuk aplikasi web berbasis .NET).

Transaction Processing Middleware: Middleware ini digunakan untuk mengelola transaksi dalam sistem terdistribusi yang memerlukan konsistensi dan integritas data. Contoh middleware ini termasuk IBM CICS dan Microsoft Distributed Transaction Coordinator (MSDTC).

Enterprise Service Bus (ESB): ESB adalah jenis middleware yang digunakan untuk mengintegrasikan aplikasi dan layanan dalam lingkungan perusahaan. Contoh ESB termasuk Apache ServiceMix dan Mule ESB.

Apa saja tantangan dalam Sistem Terdistribusi?

Komunikasi dalam sistem terdistribusi menghadapi sejumlah tantangan yang harus diatasi untuk memastikan kinerja yang baik dan keamanan. Beberapa tantangan utama dalam komunikasi terdistribusi meliputi:

Latensi: Latensi adalah keterlambatan dalam pengiriman data dari satu node ke node lain dalam jaringan. Tantangan ini bisa sangat kritis dalam aplikasi real-time seperti game online atau telemedisin.

Keamanan: Mengamankan data yang dikirim melalui jaringan adalah tantangan utama. Ini mencakup enkripsi data, otentikasi pengguna, manajemen hak akses, dan perlindungan terhadap serangan siber.

Konsistensi Data: Memastikan konsistensi data di seluruh sistem terdistribusi adalah tantangan. Terdapat perbedaan antara data yang disimpan di beberapa node, dan perubahan harus disinkronkan dengan benar.

Integrasi Aplikasi: Ketika berbagai aplikasi beroperasi dalam sistem terdistribusi, mereka perlu berkomunikasi dan berbagi data. Menjaga konsistensi dan kompatibilitas antar aplikasi ini bisa rumit.

Ketersediaan: Sistem terdistribusi harus tetap tersedia bahkan ketika ada gangguan atau kegagalan dalam komponen-komponen individualnya. Ini melibatkan manajemen dan pemulihan dari kegagalan.

Skalabilitas: Tantangan lain adalah mengelola pertumbuhan sistem. Ketika sistem perlu diubah atau ditingkatkan, itu harus dilakukan tanpa mengganggu operasi yang ada.

Manajemen Jaringan: Jaringan yang luas dengan banyak perangkat yang berbeda bisa rumit untuk dikelola. Tantangan meliputi pemantauan kesehatan jaringan, manajemen lalu lintas, dan pengoptimalan performa.

Kompatibilitas Perangkat Lunak: Aplikasi yang berbeda mungkin dikembangkan menggunakan teknologi dan bahasa yang berbeda. Integrasi dan kompatibilitas perangkat lunak ini bisa menjadi tantangan.

Beban Kerja yang Tidak Merata: Dalam sistem terdistribusi yang besar, beban kerja bisa tidak merata di antara berbagai node, yang dapat menyebabkan overutilisasi atau underutilisasi sumber daya.

Manajemen dan Monitoring: Memonitor dan mengelola performa sistem terdistribusi, mendiagnosis masalah, dan melacak penggunaan sumber daya adalah tantangan yang kompleks dalam skala yang besar.

Menyadari dan mengatasi tantangan-tantangan ini adalah kunci dalam merancang, mengimplementasikan, dan mengoperasikan sistem terdistribusi yang handal dan efisien.