IF3230
Sistem Paralel dan Terdistribusi
Intro Sistem Terdistribusi

Achmad Imam Kistijantoro (imam@informatika.org)
Judhi Santoso (judhi@informatika.org)
Anggrahita Bayu Sasmita (bayu.anggrahita@informatika.org)
Maret 2022

Apakah Sistem Tersebar?

- *"A distributed system is one in which the failure of a computer you did not even know existed can render your own computer unusable."*
  - *Leslie Lamport.*

THALES

Rockwell Collins
Building trust every day

Honeywell


ZOO

- Sistem tersebar adalah sistem yang terdiri atas komponen yang independen, namun menyediakan kesatuan layanan kepada penggunanya.

- Tanenbaum (Tanenbaum 1988) membatasi definisi sistem tersebar dengan mensyaratkan aspek transparansi terhadap pengguna sistem.

- Coulouris (Coulouris, Dollimore et al. 2001) mendefinisikan sebagai sistem yang memiliki komponen yang berada dalam satu jaringan komputer yang saling berkomunikasi hanya dengan pertukaran pesan. Karakteristik utama: independent failure, lack of global clock dan konkurensi.

- Schroeder (Schroeder 1993) mendefinisikan sederhana: beberapa komputer yang melakukan sesuatu bersama-sama, dengan tiga karakteristik: multiple computers, interconnections, shared state.

The art of solving the same problem that you can solve on a single computer using multiple computers

Apa yang ingin dicapai oleh sistem terdistribusi

Scalability - kemampuan sistem untuk tumbuh.

Performance – jumlah pekerjaan yang dapat diselesaikan sistem komputer berdasarkan waktu dan resource yang digunakan

Availability – persentase sistem berada dalam kondisi berfungsi baik

Performance

- respons time yang singkat/cepat
- throughput tinggi
- efisien, penggunaan resources yang rendah

- latency: periode antara mulai terjadinya sesuatu hingga efek kejadian tersebut dapat diamati

- Availability – downtime per tahun
  - 90%: lebih dari 1 bulan
  - 99%: kurang dari 4 hari
  - 99.9%: kurang dari 9 jam
  - 99.99%: kurang dari 1 jam
  - 99.999%: sekitar 5 menit
  - 99.9999%: sekitar 31 detik

Fault Tolerant: kemampuan sistem untuk tetap berfungsi baik meski terjadi fault

- sistem terdistribusi dibatasi oleh:
  - Jumlah node
  - Jarak antar node

- Semakin banyak jumlah node independen, semakin besar kemungkinan terjadi kegagalan
- Semakin banyak jumlah node independen, semakin banyak kebutuhan komunikasi antar node
- Semakin jauh jarak antar node (e.g. Faktor geografis), semakin lama komunikasi antar node

- Sebuah sistem memberikan jaminan layanan (SLA)
  - Jika data ditulis, seberapa cepat data tersebut bisa diakses di tempat yang berbeda
  - Setelah data ditulis, apakah akan durable?
  - Jika sistem melakukan komputasi, seberapa cepat hasil selesai dihitung
  - Jika ada komponen fail, apa dampaknya terhadap sistem?

Bagaimana cara menjamin SLA terpenuhi?

- Abstraksi: membuat sesuatu menjadi lebih mudah ditangani, dengan **menghilangkan hal2 yang tidak penting**/terkait langsung
- Model: penyederhanaan sistem sehingga hanya berisi properties yang penting

- abstraksi yang terlalu banyak penyederhanaan, membuat pemahaman lebih mudah, namun mengurangi kinerja sistem

- abstraksi dengan jaminan yang lebih **lemah**, lebih sulit dipahami, namun dapat memberi kinerja yang lebih baik

Contoh abstraksi

- Sistem diwakili dengan node/proses dan link/komunikasi

- node/proses: sesuatu yang aktif, yang dapat menjalankan program secara sekuensial
  - OS dihilangkan
  - tampilan, keyboard, hal2 yang tidak relevan dihilangkan
  - konkurensi
  - urutan eksekusi
  - kegagalan
- komunikasi
  - urutan pesan
  - jaminan pengiriman

Contoh Two Generals Problem

- 2 orang jendral sedang mengepung sebuah kota.
- mereka akan berhasil jika melakukan serangan secara serentak
- komunikasi menggunakan kurir
- kurir dapat ditangkap musuh, sehingga pesan tidak sampai

Two Generals Problem

- kemungkinan solusi
  - A: Serang saat pagi hari!
  - B: Ok, Serang saat pagi hari!
  - A: Ok, Ok, Serang saat pagi hari!
  - B: Ok, Ok, Ok, Serang saat pagi hari!
  - ...

- tidak ada solusi untuk masalah ini

Two Generals Problem

- solusi probabilistik
  - mengirim n kurir
  - jika p adalah kemungkinan sebuah kurir tertangkap, maka
  - kemungkinan semua kurir tertangkap adalah p^n

- Kemungkinan sukses diperbesar dengan menambah kurir, namun menambah biaya

Byzantine Generals Problem

- Problem:
  - n jenderal byzantine sedang mengepung kota
  - mereka harus sepakat kapan akan menyerang
  - kurir dapat dipercaya, namun ada jenderal yang mungkin berkhianat => penerima dapat mendeteksi sumber pesan
  - memerlukan mayoritas jenderal mengambil keputusan yang sama
- Kemungkinan bukan solusi
  - 5 jenderal
  - 2 loyal mengirim: serang. 2 loyal mengirim: bertahan
  - 1 pengkhianat: bilang serang ke 2 jenderal, dan bertahan ke 2 jenderal lainnya.

https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/

Byzantine Generals Problem

- Lamport, Shostak, Pease: Byzantine Fault tolerance
- diberikan n processes: dapat mentoleransi t pengkhianat jika n >= 3t+1

- Model sistem terdistribusi
  - reliable/unreliable communication
  - benign/malicious processes
  - Timing/order
- solusi bergantung pada model
  - solusi probabilistik
  - bounded solution
  - tidak ada solusi
- koordinasi antar multiple proses tidak mudah
  - unexpected failures
  - lack of synchronization

Mengapa sistem terdistribusi

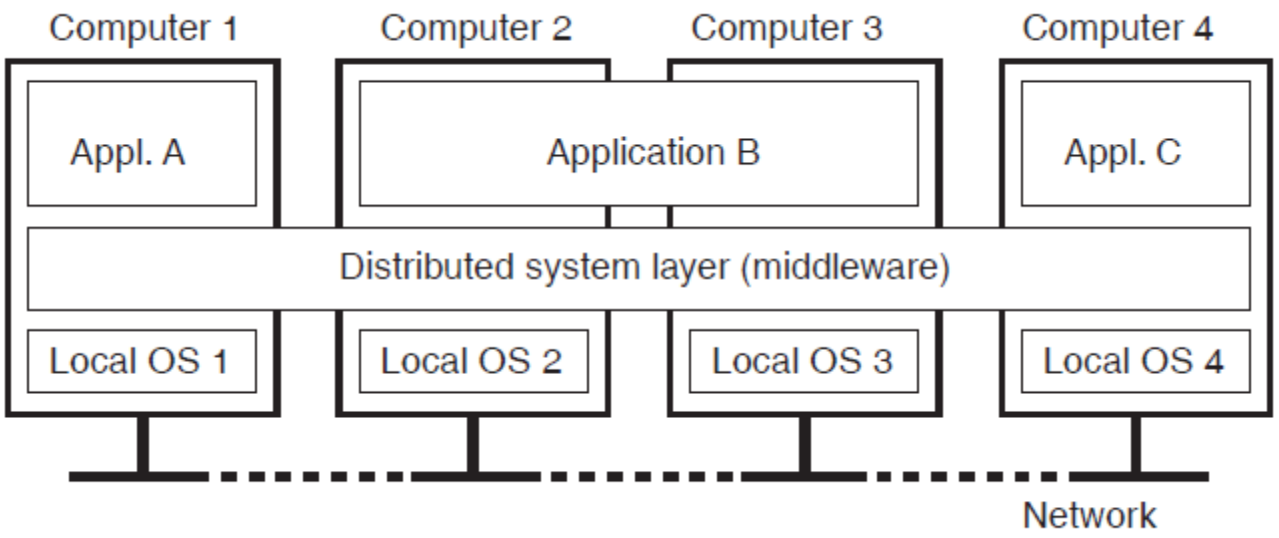
- inherent distribution
  - aplikasi yang membutuhkan sharing resources atau penyebaran informasi antar entities yang tersebar secara geografis
- Contoh
  - ATM
  - ticket reservation
  - Web
  - Games

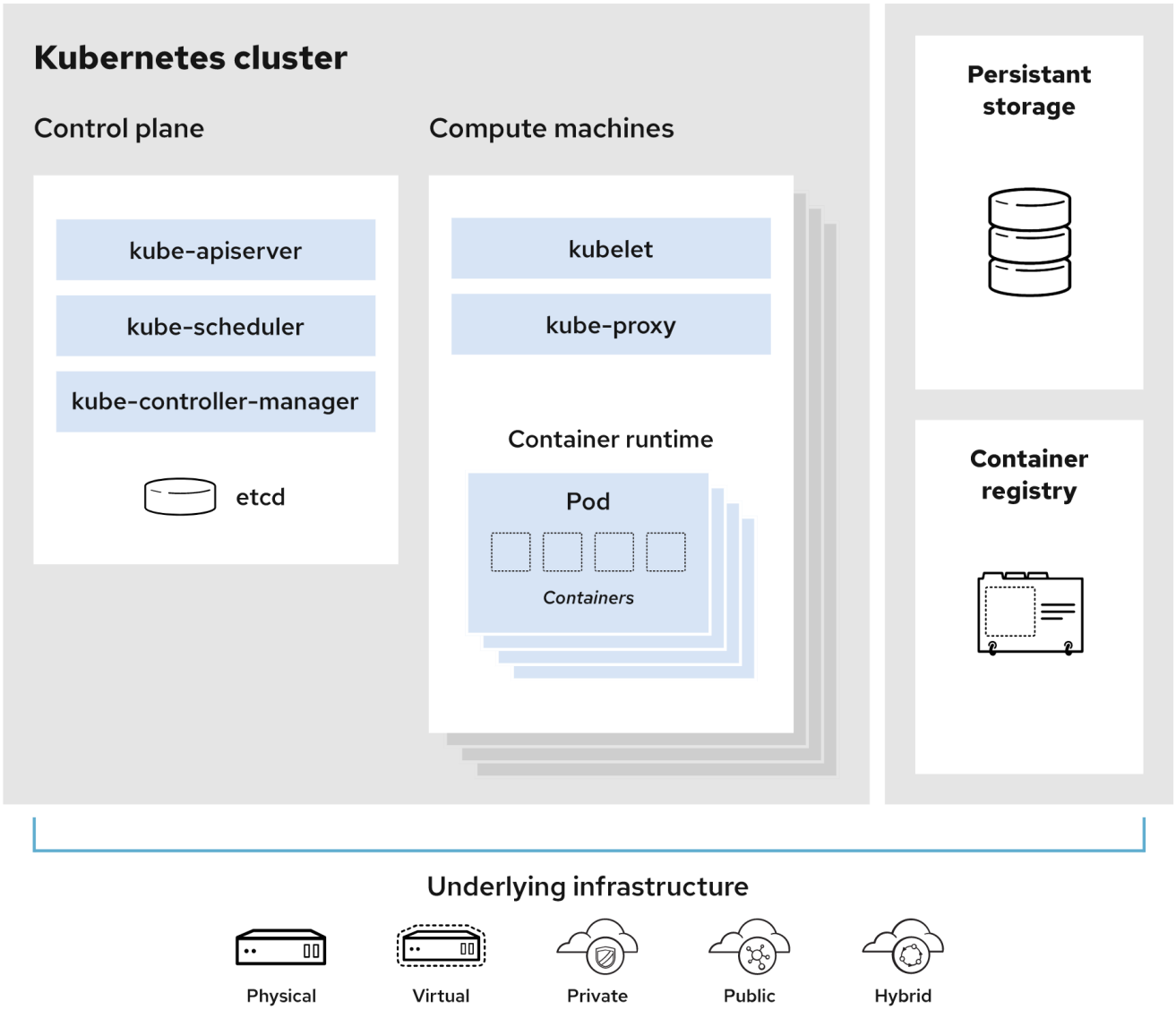
Mengapa Sistem terdistribusi

- distribusi sebagai artifak untuk solusi masalah tertentu
  - fault tolerant
  - meningkatkan performance/cost ratio
  - meningkatkan QoS
- Contoh
  - replicated servers

Mengapa sulit?

- Konkurensi
  - penanganan akses bersama
- Sinkronisasi
  - timing
- Failures
  - multiple components


Computer 1
Computer 2
Computer 3
Computer 4
Appl. A
Application B
Appl. C
Distributed system layer (middleware)
Local OS 1
Local OS 2
Local OS 3
Local OS 4
Network



https://www.redhat.com/cms/managed-files/kubernetes_diagram-v3-770x717_0.svg

Goals of distributed systems

- making resources available
- distribution transparency
- openness
- scalability

Distribution transparency

Transp.	Description
Access	Hides differences in data representation and invocation mechanisms
Location	Hides where an object resides
Migration	Hides from an object the ability of a system to change that object's location
Relocation	Hides from a client the ability of a system to change the location of an object to which the client is bound
Replication	Hides the fact that an object or its state may be replicated and that replicas reside at different locations
Concurrency	Hides the coordination of activities between objects to achieve consistency at a higher level
Failure	Hides failure and possible recovery of objects

Degree of transparency

- full distribution transparency -> berlebihan
  - user berada pada lokasi yang berbeda
  - hiding failures sepenuhnya -> impossible
    - tidak bisa membedakan antara failures dan slow
    - tidak memastikan apakah sebuah operasi berhasil dilakukan sebelum crash
  - full transparency -> cost performance

Openness

- conform ke well defined interfaces
- support portability
- easily interoperate

- bagaimana membuat sistem terdistribusi independen terhadap heterogenitas:
  - hardware
  - platform
  - language

Skalabilitas

- Istilah yang over-use

- scalability:
  - number of users/processes (size scalability)
  - max distance between nodes (geographical scalability)
  - number of administrative domains (administrative scalability)

Teknik Skalabilitas

- hide communication latency
  - asynchronous communication
  - separate handler for incoming response
  - not all applications fit this model

- distribution
  - partition data and computation across multiple machines
  - move computation to clients (javascripts, applets)
  - decentralized data (DNS)
- replication/caching
  - replicated file servers
  - web cache, web mirror
  - file cache

fallacies in developing distributed apps

- network is reliable
- network is secure
- network is homogeneous
- topology does not change
- latency is zero
- bandwidth is infinite
- transport cost is zero
- there is one administrator

Tugas/belajar mandiri

- pelajari hal yang terkait dengan 8 fallacies in distributed systems, Peter Deutch

- Network is reliable
  - Switch: MTBF: more than 50.000 hours
  - Application 24x7 mission critical

  - Power failure, human error, burst, external factor, DDOS attack

- Latency is zero
  - WAN:
  - AJAX
  - Other effects
- Bandwidth is infinite
- Network is secure
  - Internet attack: 64% annual growth (RipTech, over Fortune 1000 companies)

- Topology doesn't change
  - Topology change constantly
- There is one administrator
- Transport cost is zero
- Network is homogeneous