Sistem Terdisreibusi

Proses Thread

TKB6252 – Sistem Terdistribusi

Chalifa Chazar

www.script.id

chalifa.chazar@gmail.com

Last update : Januari 2018 | chalifa.chazar@gmail.com

Pendahuluan

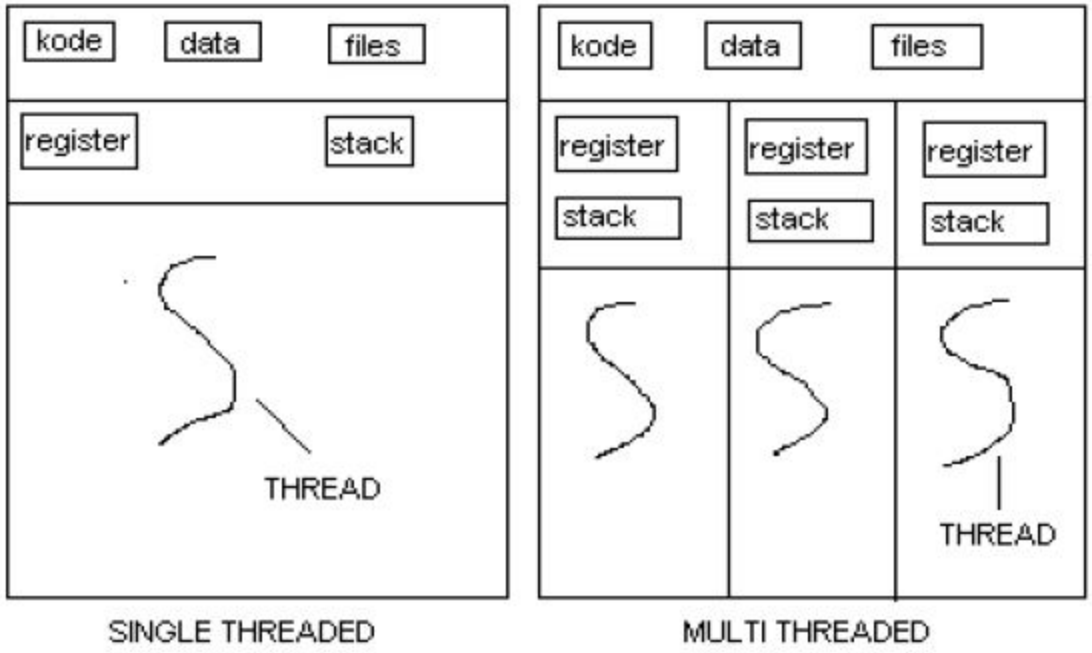
- Terdapat 2 jenis proses, yaitu proses berat/proses tradisional (heavyweight process) dan proses ringan (lightweight process)
- Dalam sistem terdistribusi thread tidak lepas dari RPC
- Dengan adanya RPC, kerja sistem menjadi lebih ringan karena efisiensi traffic komunikasi menjadi lebih tinggi

Thread

- Thread atau dikenal juga sebagai proses ringan (lightweight process) adalah sebuah alur kontrol dari sebuah proses
- Model proses yang digunakan saat ini menunjukan suatu proses program yang menjalankan eksekusi thread tunggal
- Sebuah proses tradisional (heavyweight process) mempunyai thread tunggal yang berfungsi sebagai pengendali
- Dalam sebuah thread terdapat ID tread, program counter, register, dan stack yang saling berbagi dengan thread lain dalam proses yang sama

- Pada sistem terdistribusi thread yang digunakan adalah multithread karena dalam sistem terdistribusi berhubungan dengan banyak komputer
- Thread akan menerima dan mengirim beberapa pesan task secara kontinyu asinkron
- Contoh:
  - Sebuah daemon dapat memiliki banyak thread yang menunggu untuk menerima request pesan
  - Pesan tersebut diterima oleh thread dan diproses secara bersamaan dengan thread lainnya
  - Tugas kernel yang akan meng-create thread baru bila ada request datang

- Perbedaan antara proses dengan thread tunggal dengan proses dengan multithread adalah proses dengan thread yang banyak dapat mengerjakan lebih dari satu tugas pada satu satuan waktu



fa.chazar@gmail.com

Keuntungan Multithread

- Tanggap

  Multithread mengijinkan program untuk berjalan terus walaupun bagian program tersebut di-blok atau sedang mengerjakan operasi yang lama atau panjang
- Pembagian sumber daya

  Thread akan membagi sumber daya proses. Keuntungannya aplikasi memiliki aktifitas thread dengan alokasi memori yang sama
- Ekonomis

  Kemampuan thread yang dapat membagi sumber daya dan memori akan membuat alokasi sumber daya dan memori dapat ditekan
- Pemberdayaan arsitektur multiprosessor

  Arsitektur multiprosessor memiliki keuntungan dimana tiap thread yang dibentuk dapat berjalan secara parallel pada prosesor yang berbeda

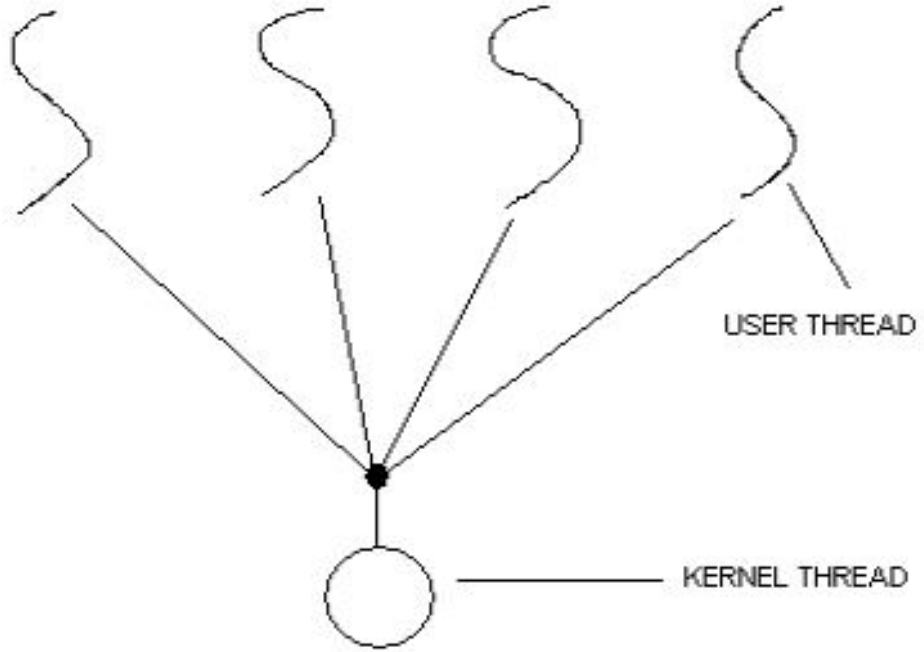
Multithread Model

- Many-to-One Model
- One-to-One Model
- Many-to-Many Model

Many-to-One Model

- Memetakan banyak user-level thread ke satu kernel thread.
- Pengaturan thread dilakukan di user space, oleh karena itu model ini efisien, tetapi mempunyai kelemahan yang sama dengan user thread
- Selain itu karena hanya satu thread yang bisa mengakses thread pada suatu waktu maka multiple thread tidak bisa berjalan secara paralel pada multiprocessor
- User-level thread yang diimplementasi pada sistem operasi yang tidak mendukung kernel thread menggunakan Many-to-One model

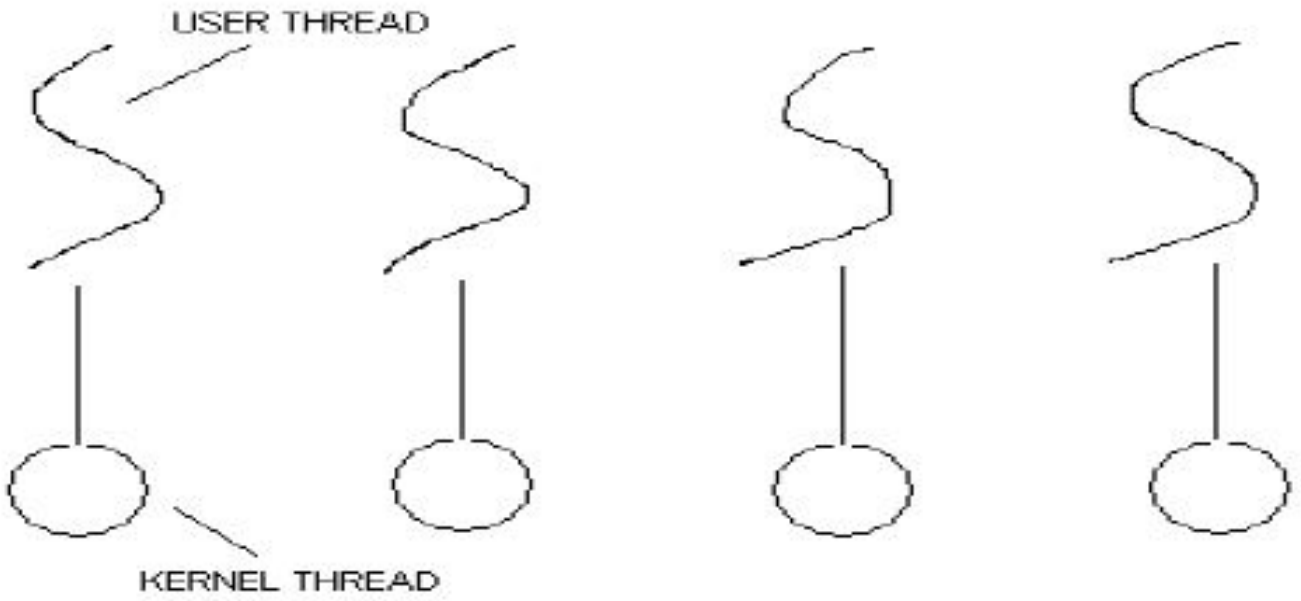
Many-to-One Model



One-to-One Model

- Memetakan setiap user thread ke kernel thread
- Ia menyediakan lebih banyak concurrency dibandingkan Many-to-One model
- Kelemahannya model ini adalah setiap pembuatan user thread membutuhkan pembuatan kernel thread
- Karena pembuatan thread bisa menurunkan performa dari sebuah aplikasi maka implmentasi dari model ini membatasi jumlah thread yang dibatasi oleh sistem

One-to-One Model



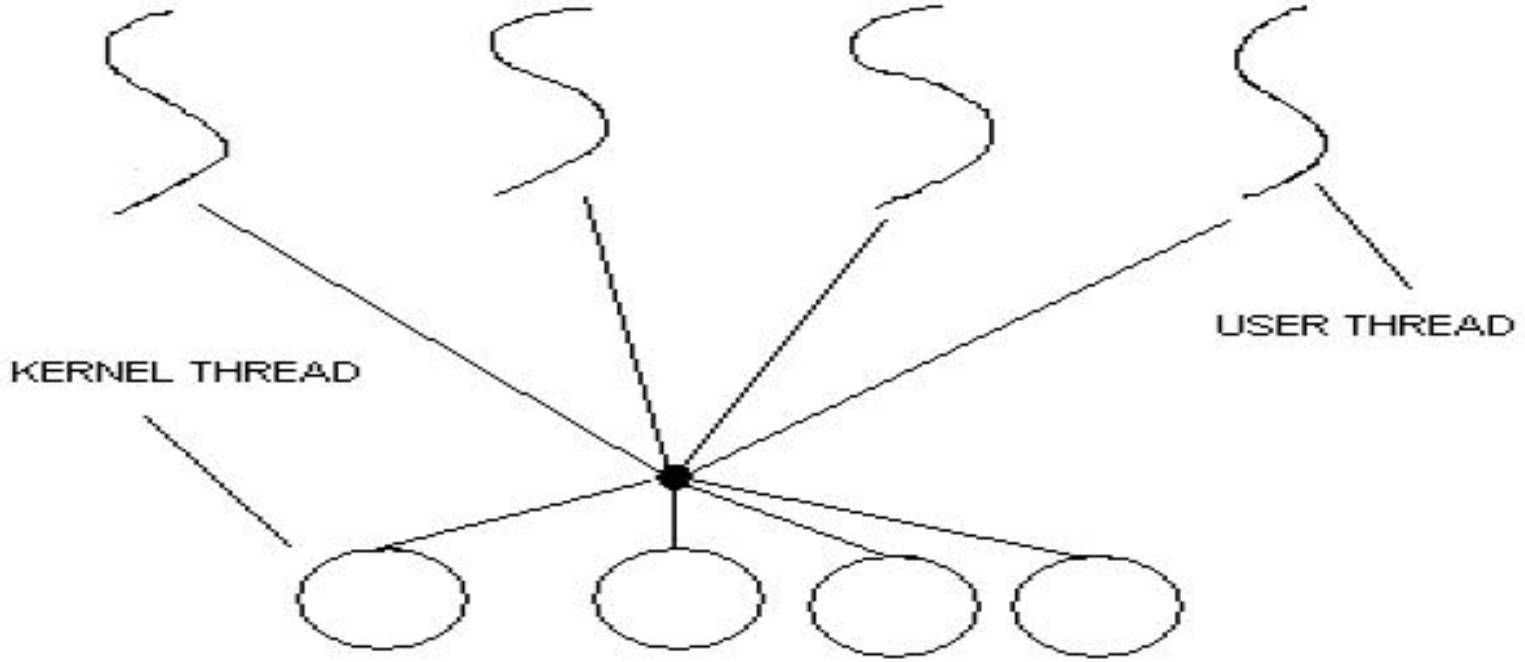
Many-to-Many Model

- Memetakan banyak user-level thread ke kernel thread yang jumlahnya lebih kecil atau sama banyaknya dengan user-level thread
- Jumlah kernel thread bisa spesifik untuk sebagian aplikasi atau sebagian mesin
- Pada Many-to-One model, developer diijinkan untuk membuat user thread sebanyak yang ia mau tetapi concurrency tidak dapat diperoleh karena hanya satu thread yang bisa dijadwal oleh kernel pada suatu waktu
- One-to-One menghasilkan concurrency yang lebih tetapi developer harus hati-hati untuk tidak menciptakan terlalu banyak thread dalam suatu aplikasi (dalam beberapa hal, developer hanya bisa membuat thread dalam jumlah yang terbatas)

Many-to-Many Model

- Many-to-Many model memiliki keunggulan dibandingkan 2 model sebelumnya yaitu:
  - Developer bisa membuat user thread sebanyak yang diperlukan
  - Kernel thread yang bersangkutan bisa bejalan secara pararel pada multiprocessor
  - Juga ketika suatu thread menjalankan blocking system call maka kernel dapat menjadwalkan thread lain untuk melakukan eksekusi.

Many-to-Many Model



Cancellation

- Thread cancellation adalah tugas untuk memberhentikan thread sebelum ia menyelesaikan tugasnya
- Thread yang akan diberhentikan biasa disebut dengan target thread

2 Cara Cancellation Thread

- Pemberhentian target bisa dilakukan dengan 2 cara, yaitu:
  - Asynchronous cancellation : suatu thread seketika itu juga memberhentikan target thread
  - Defered cancellation : target thread secara periodik memeriksa apakah dia harus berhenti, cara ini memperbolehkan target thread untuk memberhentikan dirinya secara terurut

Asynchronous cancellation

- Hal yang sulit dari pemberhentian thread ini adalah ketika terjadi situasi dimana sumber daya sudah dialokasikan untuk thread yang akan diberhentikan
- Selain itu kesulitan lain adalah ketika thread yang diberhentikan sedang meng-update data yang ia bagi dengan thread lain
- Sistem operasi akan mengambil kembali sumber daya dari thread yang diberhentikan tetapi seringkali sistem operasi tidak mengambil kembali semua sumber daya dari thread yang diberhentikan.

Defered cancellation

- Cara kerja dari deffered cancellation adalah dengan menggunakan 1 thread yang berfungsi sebagai pengindikasi bahwa target thread hendak diberhentikan
- Tetapi pemberhentian hanya akan terjadi jika target thread memeriksa apakah ia harus berhenti atau tidak
- Hal ini memperbolehkan thread untuk memeriksa apakah ia harus berhenti pada waktu dimana ia bisa diberhentikan secara aman yang aman
- Thread tersebut dikenal sebagai cancellation points .

</TERIMA KASIH>

Chalifa Chazar, S.T, M.T

Email: chalifa.chazar@gmail.com

script.id