



Komputasi Paralel Sinkronisasi

Ketua : Huda Izzatul Haq

Anggota :

- Ardith Lutfiawan
- Ahmad Fathan Syakir
- Muhammad Azhar Rasyad

Sekolah Tinggi Teknologi Terpadu
Nurul Fikri

Teknik Informatika 2017

Sinkronisasi Proses

Sinkronisasi Proses berarti berbagi sumber daya sistem dengan proses sedemikian rupa sehingga, akses bersamaan ke data ditangani sehingga meminimalkan kemungkinan data yang tidak konsisten.

Mempertahankan konsistensi data menuntut mekanisme untuk memastikan pelaksanaan proses kerja sama yang tersinkronisasi. Sinkronisasi Proses diperkenalkan untuk menangani masalah yang muncul saat beberapa proses dieksekusi [0].

Sinkronisasi Proses vs Data

Sinkronisasi proses yaitu eksekusi secara bersamaan dari multiple threads atau proses untuk mencapai kesamaan aksi sehingga mereka melakukan urutan tindakan tertentu. Kunci, mutex, dan semaphore adalah contoh sinkronisasi proses.

Sinkronisasi data yaitu melibatkan pemeliharaan data untuk menjaga banyak salinan data yang koheren satu sama lain, atau untuk menjaga integritas data. Misalnya, replikasi basis data digunakan untuk menjaga banyak salinan data disinkronkan dengan server basis data yang menyimpan data di lokasi yang berbeda [1].

Fungsi Sinkronisasi

Sinkronisasi dirancang agar kooperatif, menuntut agar setiap utas mengikuti mekanisme sinkronisasi sebelum mengakses sumber daya yang dilindungi untuk hasil yang konsisten.

Mengunci, memberi sinyal, tipe sinkronisasi ringan, pemintalan, dan operasi yang saling terkait adalah mekanisme yang terkait dengan sinkronisasi dalam .NET [1].

Waktu dilakukan Sinkronisasi

- *Critical Section Problem*
- *Race Condition [2]*

Waktu 1 *Critical Section Problem*

Critical Section adalah segmen kode yang mengakses variabel bersama dan harus dijalankan sebagai aksi atom. Ini berarti bahwa dalam sekelompok proses kerja sama, pada titik waktu tertentu, hanya satu proses yang harus menjalankan bagian kritisnya.

Jika ada proses lain yang juga ingin menjalankan bagian kritisnya, ia harus menunggu sampai yang pertama selesai [2].

Waktu 2 *Race Condition*

Ketika lebih dari satu proses mengeksekusi kode yang sama atau mengakses memori yang sama atau variabel apa pun yang dibagi dalam kondisi itu, ada kemungkinan bahwa output atau nilai variabel yang dibagikan salah sehingga untuk semua proses yang melakukan perlombaan mengatakan bahwa saya output benar kondisi ini dikenal sebagai kondisi balapan.

Beberapa proses mengakses dan memproses manipulasi data yang sama secara bersamaan, maka hasilnya tergantung pada urutan tertentu di mana akses terjadi [2].

Sinkronisasi dalam *OpenMP*

Sinkronisasi merupakan hal yang perlu diperhatikan. bagaimana tidak? jika kita telah membagi-bagi pekerjaan ke beberapa thread / processor, hal terakhir yang harus kita lakukan yaitu menyatukan data-data hasil proses yang telah dikerjakan masing-masing processor.

Ada tiga jenis sinkronisasi pada *OpenMP* :

- Barrier
- Atomic
- Critical

Sinkronisasi *OpenMP Barrier*

Barrier yaitu Direktif untuk membuat batas, thread akan menunggu sampai semua thread sampai pada batas ini [3].

Sinkronisasi *OpenMP Critical*

Critical yaitu Direktif untuk membuat wilayah kritis dalam wilayah paralel, hanya satu thread dalam satu waktu yang bisa masuk ke wilayah kritis ini. Wilayah kritis perlu dibuat untuk melindungi shared variable dari race condition yang mengakibatkan inkonsistensi data [3].

Sinkronisasi *OpenMP Atomic*

Atomic Direktif yang fungsinya sama seperti *critical*, tetapi hanya berlaku untuk satu statement aritmatika saja [3].

Sinkronisasi dalam *OpenMP*

```
#include <stdio.h>
#include <omp.h>
#define N 1200
#define T 4

int main() {
    int sum = 0;
    #pragma omp parallel num_threads(T)
    {
        int i;
        for (i = 0; i < N/T; i++)
            #pragma omp atomic
            sum++;           // critical section
    }
    printf("%d\n", sum);

    return 0;
}
```

Daftar Pustaka

- [0] <https://www.studytonight.com/operating-system/process-synchronization>
- [1] <https://www.techopedia.com/definition/13390/synchronization-dot-net>
- [2] <https://www.geeksforgeeks.org/introduction-of-process-synchronization/>
- [3] <http://semangatkecil.blogspot.com/2015/12/sinkronasi-pada-openmp-parallel.html>