

DESAIN DAN IMPLEMENTASI KOMPUTASI PARALEL PADA EXCEL

Agus Kurniawan, Sagi Arsyad

Fakultas Ilmu Komputer,

Universitas Indonesia,

Depok, Indonesia, 16424

E-mail: saar50@ui.ac.id , agus@cs.ui.ac.id

ABSTRAKS

Microsoft Excel dikenal luas sebagai aplikasi bisnis, namun karena keterbatasan kemampuan hardware, Excel tidak memiliki kemampuan komputasi untuk data yang sangat besar. Pada makalah ini, kami menghadirkan desain dan implementasi Excel untuk dapat memanfaatkan kemampuan komputasi yang sangat besar dengan memanfaatkan HPC Server.

Kata Kunci: hpc, excel add-in, parallel computing, mpi

1 PENDAHULUAN

1.1. Latar Belakang

Microsoft Excel dikenal luas sebagai aplikasi bisnis untuk mengolah *spreadsheet*. Namun karena keterbatasan kemampuan kalkulasi PC dimana Excel berada, kita tidak dapat melakukan kalkulasi dalam jumlah yang sangat besar.

High Performance Computing (HPC) dikenal memiliki kemampuan komputasi yang sangat tinggi. HPC mampu memanfaatkan beberapa computer untuk menjalankan aplikasi paralel dengan cepat dan efisien.

Di sisi lain, HPC tidak memiliki *user interface* yang bersahabat dan membutuhkan biaya yang besar untuk membangun infrastruktur HPC. Sehingga, pengguna lebih memilih menggunakan PC mereka untuk melakukan kalkulasi Excel walaupun untuk menghitung data yang sangat besar, akan membutuhkan waktu yang lama dan kadang kala membuat sistem tidak responsif.

1.2. Tujuan

Tujuan penelitian ini adalah membangun sistem *middleware* yang menjembatani antara aplikasi Excel dan HPC server untuk melakukan komputasi yang kompleks.

Keuntungan bagi pengguna adalah dapat mempersingkat jangka waktu yang dibutuhkan untuk mengolah data yang sangat besar tanpa harus memikirkan HPC server yang bertidak dibalik layar.

1.3. Ruang Lingkup

Batasan pada desain dan implementasi yang dilakukan pada riset ini adalah :

- Kalkulasi yang dilakukan hanya kalkulasi sederhana, hanya untuk membuktikan kemampuan kalkulasi secara paralel
- HPC server masih menggunakan hampir sepenuhnya teknologi Microsoft untuk mempermudah pengembangan

2. LANDASAN TEORI

2.1. Message Parsing Interface

Message Parsing Interface (MPI) merupakan spesifikasi *library* untuk *message parsing* yang diusulkan sebagai standar oleh sebuah komite yang terdiri dari *vendor*, pelaku implementasi dan pengguna.

2.2. MPI.NET

MPI.NET merupakan implementasi MPI untuk .NET framework environment. MPI.NET menyediakan dukungan untuk pengembangan MPI menggunakan semua bahasa yang didukung .NET Framework, juga mempermudah pengembangan aplikasi paralel yang berjalan di cluster.

2.3 Microsoft HPC Server 2008

Microsoft HPC Server 2008 merupakan implementasi HPC pada platform Microsoft Windows Server. Salah satu fiturnya antara lain integrasi dengan *cluster environment*, termasuk *job scheduler*, Message Parsing Interface v2 (MPI2), *cluster management* dan *monitoring component*.

2.4 .NET Framework

.NET Framework terdiri dari *runtime environment* dan kumpulan class *library* untuk membangun software pada platform Microsoft Windows, termasuk Windows Communication Foundation dan Excel add-in.

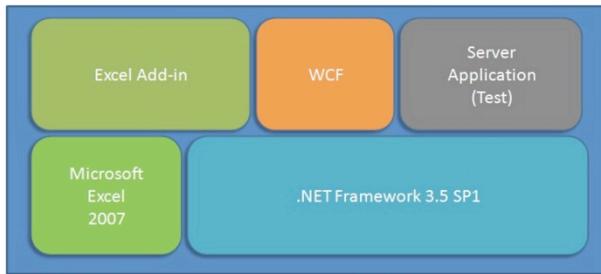
2.5. Windows Communication Foundation

Windows Communication Foundation (WCF) merupakan bagian dari .NET framework, berfungsi untuk menyediakan *unified programming model* dan *service oriented application* (SOA). WCF dirancang mampu dengan interop dengan platform lainnya.

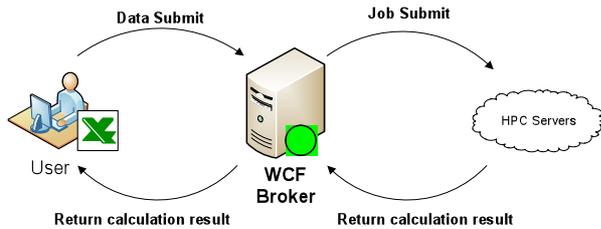
3. IMPLEMENTASI

3.1. High Level Logical Design

Ide utama dari implementasi ini adalah bagaimana Excel dapat memanfaatkan kemampuan komputasi dari HPC server. Beberapa element yang membentuk sistem ini dapat dilihat pada gambar 1.



Gambar 1. Framework stack



Gambar 2. Mekanisme perhitungan data

3.1.1. Excel HPC Add-in

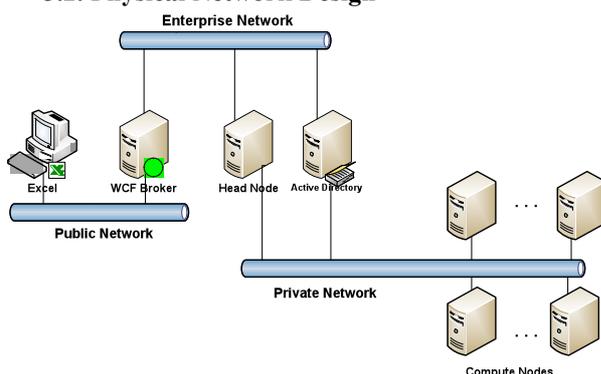
Excel HPC add-in merupakan custom extension untuk Excel yang kami bangun .NET Framework. Excel add-in bekerja sebagai bagian dari *Ribbon Toolbar* Microsoft Excel pada PC pengguna. Framework stack untuk Excel add-in ini diilustrasikan pada gambar 2.

Excel HPC ini berguna mengumpulkan data yang diinput oleh pengguna dan mengirim data tersebut ke WCF Broker.

3.1.2. WCF Broker

WCF Broker bertindak sebagai penerima data yang dikirimkan oleh Excel. setelah itu, dilakukan translasi data dan melakukan *job submit* ke Head Node.

3.2. Physical Network Design



Gambar 3. Arsitektur fisik jaringan

Seperti yang ditunjukkan pada gambar 3, pada level fisik sistem Excel HPC terdapat 3 jaringan, yaitu:

- Public Network.** jaringan yang menghubungkan antara para pengguna dan WCF broker
- Enterprise Network.** jaringan internal HPC Server yang digunakan WCF Broker untuk melakukan job submit ke Head Node

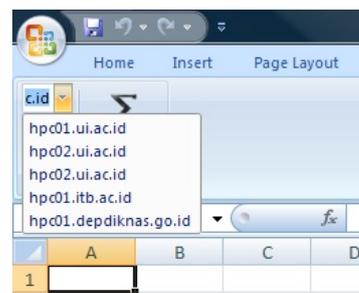
- Private Network.** jaringan internal antara head node dan kumpulan compute nodes

Pada level fisik ini juga, entiti yang memiliki peran antara lain :

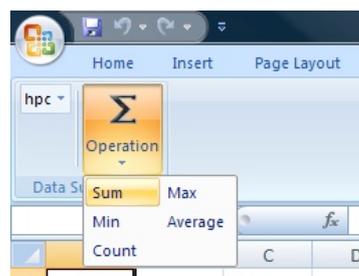
- Client PC (Excel).** *Workstation* yang dipakai pengguna untuk input data pada Excel. Entity ini merupakan host bagi aplikasi Microsoft Excel dan Excel add-in
- WCF Broker.** Entity ini menjadi host bagi WCF *service* dan juga sebagai penghubung HPC server dan client. Selain itu, entity ini bertugas melakukan *job submit* ke Head node.
- Active Directory.** Entity ini bertindak mengatur hak akses pengguna maupun komputer dalam pada jaringan enterprise network dan private network
- Head Node.** Entity ini bertugas melakukan pembagian job kepada nodes yang sudah dialokasikan
- Compute Node.** Entity ini bertindak menerima *job* yang telah dibagikan oleh Head node.

3.3. Langkah Komputasi

Pada langkah pertama, pengguna memilih *cell* pada Excel yang akan dilakukan kalkulasi. Kemudian memilih HPC server yang akan melakukan komputasi dan fungsi yang akan dijalankan. Pemilihan daftar HPC server yang bisa dipakai dan fungsi yang akan dijalankan ini bisa dilakukan melalui *ribbon toolbar* pada Excel seperti yang diperlihatkan pada gambar 5 dan gambar 6.



Gambar 5. Pengguna melakukan pemilihan HPC server



Gambar 6. Pengguna melakukan fungsi kalkulasi

Pada saat pengguna melakukan *submit*, Excel HPC add-in melakukan pengumpulan data cell yang dipilih

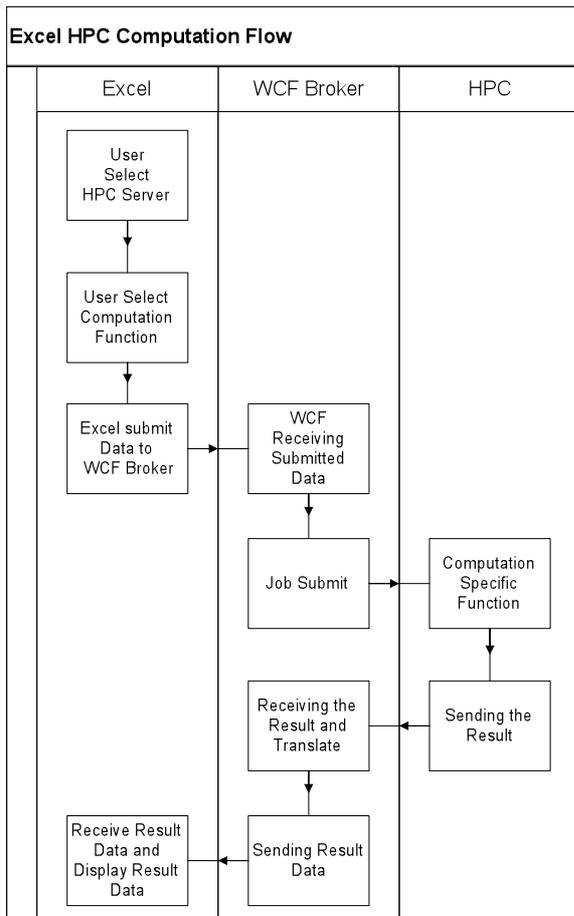
oleh pengguna dan mengirimnya dalam bentuk array ke WCF Broker. Data yang diterima oleh WCF, dijadikan parameter pada saat *job submit* ke Head Node yang selanjutnya dilakukan kalkulasi pada HPC Server.

Head Node yang menerima *job*, melakukan pembagian *job* kalkulasi kepada sejumlah compute node yang telah dialokasikan yang selanjutnya dilakukan kalkulasi pada masing-masing compute node.

Data hasil kalkulasi dikembalikan dari Head Node ke WCF broker yang selanjutnya ditranslasikan dan dikirim kembali ke Excel. Ilustrasi dari langkah-langkah diatas, dapat dilihat pada gambar 4.

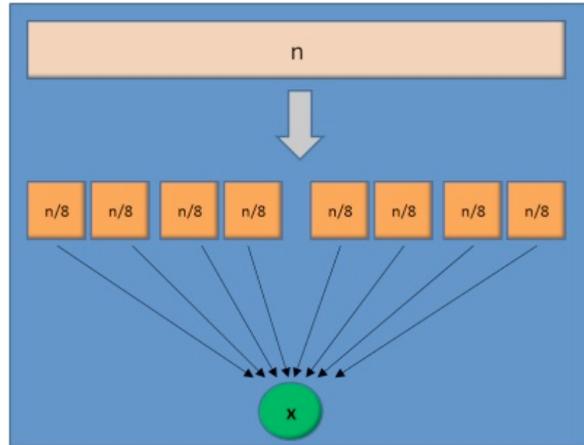
3.4. Test Bed

Untuk pengujian komputasi paralel pada HPC server, digunakan perhitungan sederhana. Yaitu menghitung nilai jumlah total (sum) dan nilai rata-rata (average) dari seluruh elemen set bilangan desimal.



Gambar 4. Langkah komputasi keseluruhan

3.4.1. Perhitungan nilai jumlah total



Gambar 5. Ilustrasi pembagian pekerjaan untuk n array ke beberapa nodes

Pengujian dilakukan dengan memecah perhitungan n jumlah elemen array ke 8 unit compute node seperti ilustrasi gambar 5. Masing-masing nodes akan menghitung penjumlahan sebanyak n/8 element array yang berbeda. Penjabaran operasi kalkulasi ini dapat digambarkan formula sebagai berikut:

$$\sum_{i=0}^n x_i = \sum_{j=0}^y \left(\sum_{i=j \lfloor \frac{n}{y} \rfloor}^{(j \lfloor \frac{n}{y} \rfloor) - 1} x_i \right)$$

dimana:

- n = jumlah elemen array
- y = jumlah nodes yang dialokasikan

misalnya kita memiliki 1000 elemen array dan akan dilakukan kalkulasi pada 8 node. Maka, node 1 akan melakukan perhitungan penjumlahan untuk elemen array ke 0 sampai dengan 124, node 2 akan melakukan perhitungan penjumlahan elemen array ke 125 sampai dengan 249, dan seterusnya hingga node kedelapan. Data hasil kalkulasi dari masing-masing nodes, dikembalikan ke Head node. Di Head node, kembali dilakukan perhitungan jumlahan nilai hasil kalkulasi dari Node 1 sampai dengan Node 8 untuk mendapatkan nilai total akhir.

3.4.2. Perhitungan nilai rata-rata total

Seperti halnya pada perhitungan nilai jumlah total pada sub bagian 3.4.1, kami melakukan pengujian perhitungan nilai rata-rata dari sebuah arran sejumlah n elemen ke 8 unit compute node seperti yang diilustrasi kan gambar 5. Masing-masing nodes melakukan perhitungan nilai rata-rata untuk n/8 buah elemen array yang berbeda. Penjabaran operasi kalkulasi ini dapat digambarkan formula sebagai berikut:

$$avg(x_0 \dots x_n) = avg \left(avg \left(x_0 \dots x_{\lfloor \frac{n}{y} \rfloor - 1} \right) \dots avg \left(x_{(j \lfloor \frac{n}{y} \rfloor) - 1} \dots x_n \right) \right)$$

dimana:

- n = jumlah elemen array

y = jumlah nodes yang dialokasikan

misalnya kita memiliki 1000 elemen array dan akan dilakukan kalkulasi pada 8 node. Maka, node 1 akan melakukan perhitungan nilai rata-rata untuk elemen array ke 0 sampai dengan 124, node 2 akan melakukan perhitungan rata-rata elemen array ke 125 sampai dengan 249, dan seterusnya hingga node kedelapan. Data hasil kalkulasi dari masing-masing nodes, dikembalikan ke Head node. Di Head node, kembali dilakukan perhitungan nilai rata-rata nilai hasil kalkulasi dari Node 1 sampai dengan Node 8 untuk mendapatkan nilai total akhir.

3.4.3. Hasil analisa perhitungan paralel

Kalkulasi dengan menggunakan proses perhitungan secara paralel seperti diatas, dengan mengabaikan waktu yang dibutuhkan untuk transfer data antar entity, maka dapat diperkirakan waktu yang dibutuhkan untuk melakukan kalkulasi pada x buah nodes adalah :

$$T = \frac{t_n}{x} + t_x$$

dimana:

T = Total waktu pada perhitungan paralel

x = jumlah nodes yang dialokasikan

t_n = waktu yang dibutuhkan untuk perhitungan secara linear

t_x = waktu yang dibutuhkan untuk menghitung array sebanyak x elemen

4. KESIMPULAN

Memfaatkan kemampuan komputasi HPC untuk melakukan perhitungan data Excel. Hal ini merupakan suatu solusi yang cukup berguna bagi sebuah instansi yang memiliki ratusan atau ribuan karyawan yang mana masing-masing dari mereka melakukan kalkulasi data dalam jumlah yang sangat besar.

Namun di sisi lain pengadaan infrastruktur HPC membutuhkan biaya yang cukup besar, sehingga implementasi Excel HPC ini disarankan hanya untuk instansi berskala besar.

5. KELANJUTAN PENELITIAN

Penelitian ini dapat dilanjutkan dengan implementasi HPC pada platform selain Microsoft Windows.

6. UCAPAN TERIMA KASIH

Pada kesempatan ini, saya ingin mengucapkan terima kasih kepada Agus Kurniawan atas bantuan dan bimbingannya dalam pengerjaan riset ini.

PUSTAKA

- ANL Mathematics and Computer Science.** *Message Passing Interface*. [Online] [Cited: April 14, 2010]. <http://www.mcs.anl.gov/research/projects/mpi/>
- Indiana University.** *MPI.NET: High-Performance C# Library for Message Passing*. [Online] [Cited: April 15, 2010]. <http://osl.iu.edu/research/mpi/net/>
- Microsoft Corporation** (September, 2008). *Technical Overview of Windows HPC Server 2008*. Microsoft Corporation.
- Microsoft Corporation.** *.NET Framework Conceptual Overview*. Microsoft Corporation. [Online] [Cited: April 16, 2010]. <http://msdn.microsoft.com/library/zw4w595w.aspx>
- Microsoft Corporation.** *Windows Communication Foundation*. Microsoft Corporation. [Online] [Cited: April 16, 2010]. <http://msdn.microsoft.com/en-us/netframework/aa663324.aspx>
- David Chappel, Chappell & Associates** (September, 2007). *Introducing Windows Communication Foundation*. Microsoft Corporation.
- Microsoft Corporation.** *Windows Server 2008 R2: Active Directory*. [Online] [Cited: April 17, 2010]. <http://www.microsoft.com/windowsserver2008/en/us/ad-main.aspx>