July 2005

EXPERIMENTAL VALIDATION OF THE GRID SPEEDUP THEORY

Master of Science Thesis

Yonggang Zhang

Supervisor: **Dr. Alfons G. Hoekstra**

Mentor: **Prof. Dr. Chris Jesshope**

Graduation Professor: **Prof. Dr. Peter.M.A Sloot**

A thesis submitted to the Section Grid Computing Science at University of Amsterdam in partial fulfillment of the requirements of the degree Master of Science in Grid Computing Science.

ABSTRACT

The traditional speedup theory is not very "new "concept in HPC (High Performance Computing) community. However, the traditional theory has not been suitable and sufficient for Grid Computing. At present, the Grid computing community has started with challenges to demonstrate larger grid speedups on a real Grid computing environment. Unfortunately, there is no any reliable metric available yet.

A.G. Hoekstra [Hoekstra05] made an extension on the traditional speedup theory and introduced the concept of Grid Speedup and then they analyzed it theoretically on a Homogeneous Computational Grid. To the best of our knowledge, this is the first study of Grid Speedup in HPC community.

In this project, a series of experiments are set up to verify the Grid Speedup Theory using several applications on grid DAS-2.

ACKNOWLEDGEMENTS

I am indebted to my supervisor, mentors, friends, classmates and institutions as well as family members in completing my thesis. First and foremost, my sincere gratitude goes to my supervisors Dr. Alfons G. Hoekstra and Prof. Dr. Peter M.A. Sloot for their guidance, enthusiasm and an extremely helpful and generous nature.

I would like to thank my mentor Prof. Dr. Chris Jesshope, Dr. Jaap Kaandorp for taking the time to help me in choosing my thesis project.

My grateful thanks are due to staff at the institutions of DAS-2 and DutchGrid for their good helps in using the facilities.

The helps of Dr. Kamil Iskra and Drs. Jiangjun Cui are really appreciated when I was confused with some questions.

I also thank my friends and classmates, who often encouraged me when I was lacking in confidence during my study.

Last, but not the least, I am indebted to my families, in particular to my wife, without whose love, efforts and sacrifice, this would not have been possible.

LIST OF TABLES

LIST OF FIGURES

CONTENTS

Page

Chapter I Introduction

At the beginning of this section, the objective and background of this project are mentioned. This section mainly reviews some concepts useful in later discussions on speedup models and some well-known laws widely used in the parallel computing community.

1.1 Project objective and Background

The objective of this project is to validate the grid speedup theory developed by Hoekstra [Hoekstra05]. Grid computing is a huge extension on the parallel computing community and it also predicts the future of HPC (**H**igh **P**erformance **C**omputing). Researchers in Parallel Computing are usually interested in the approaches to make the performance quantitative. But, there do not exist suitable metrics capable of being used in a real grid environment. Hoekstra's contributions make it possible to evaluate the gain of performance on a real grid. However, so far Grid speedup is a theoretical concept, not tested on real applications running in a computational grid. This study is meant to test the concept of Grid Speedup on a real application.

1.2 Essential Concepts

Speedup *speedup* is strictly defined as the ratio of the execution time of the best possible sequential algorithm on a single processor to the parallel execution time of the selected algorithm on a p-processor parallel system under the assumption of both algorithm used to solve the same problem. Thus, we have:

$$S(p) = \frac{T(1)}{T(p)} \qquad (1-1)$$

Relative speedup Eq.(1-1) is often called *absolute speedup*. Since it is difficult to find the "best" sequential algorithm to solve a problem, the concept of *relative speedup* has to be introduced in most cases. The execution time of the parallel program will approximate to $T(1)$ if the parallel program runs on a single processor. Because $T(1)$

and $T(p)$ are both based on the same parallel program, the ratio of $T(1)$ to $T(p)$ is called *relative speedup*. In the following discussions, *speedup* is assumed to have the meaning of *relative speedup* unless stated otherwise.

Efficiency efficiency is defined as the ratio of the speedup to the number of processors busy in parallel computing. Or:

$$E(p) = \frac{S(p)}{p} = \frac{T(1)}{p \times T(p)} \qquad (1-2)$$

Parallelism Profile DOP (*Degree Of Parallelism*) is defined as the maximum number of processors participating in executing a program at a particular instant over time, assuming that an unbounded number of processors and system resource are available. That means that when k processors are busy during a period, we have DOP=k. To say the least, the DOP may not be always achievable in real environment since it often changes at different periods of time during the execution cycle. But the plot of the DOP over the execution tine maybe is very useful in some theoretical studies. This plot is usually called *Parallelism Profile*. Hence the average DOP is quite a relatively useful metric in studying speedup and efficiency. It is defined as the average number of processors used to execute a program. Accordingly, the average parallelism can be written as:

$$A = \frac{\left(\sum_{k=1}^{m} k \times t_k\right)}{\left(\sum_{k=1}^{m} t_k\right)} \qquad (1-3)$$

Where:

A: Average parallelism in total

m: Maximum DOP in a profile

k: DOP in a period t_k

It should be noted that in this definition communication overhead is not taken into account. Eager et al. [Eager89] investigated the relationship among the average parallelism, speedup and efficiency:

$$S(p) \geq \frac{p \times A}{p + A - 1}$$

$$\text{and} \qquad (1-4)$$

$$E(p) \geq \frac{A}{p + A - 1}$$

$S(p)$ is lower bounded by $p \times A/(p+A-1)$ and $E(p)$ by $A/(p+A-1)$. We can find that $S(p) \to p$ if $p << A$, otherwise $S(p) \to A$ if $p >> A$.

Asymptotic Speedup Assuming that W is the total amount of work of an application, W_k is the amount of work executed with $DOP = k$ and ⊚ is the *computing capacity* of each processor(homogenous system architecture), which can be approximated by execution rate, such as MIPS or Mflops [Hwang93], thus, we have:

$$W_k = k \times \Delta \times t_k$$
$$W = \left(\sum_{k=1}^{m} W_k\right) = \Delta\left(\sum_{k=1}^{m} k \times t_k\right) \qquad (1-5)$$

If communication latency and the other system overhead are temporarily not taken into consideration, the execution time of W on single processor can be calculated by:

$$T(1) = \sum_{k=1}^{m} t_k(1) = \sum_{k=1}^{m} \frac{W_k}{\Delta} \qquad (1-6)$$

And on an infinite number of processors can be computed by:

$$T(\infty) = \sum_{k=1}^{m} t_k(\infty) = \sum_{k=1}^{m} \frac{W_k}{k\Delta} \qquad (1-7)$$

We have the asymptotic speedup:

$$S_\infty = \frac{T(1)}{T(\infty)} = \frac{\sum_{k=1}^{m} W_k}{\sum_{k=1}^{m} \frac{W_k}{k}} = \sum_{k=1}^{m} k = A \qquad (1-8)$$

Thus, Eq.(1-8) is equivalent to Eq.(1-4). Under the assumption of ignoring any communication overhead and having an unbounded number of processors available, without doubt, the asymptotic speedup is the most ideal speedup that can be achievable.

1.3 Fundamental Laws

In general parallel computing, we are accustomed to evaluating the performance of our parallel programs via three traditional models: *Fixed-size Model, Fixed-time Model and Fixed-Memory Model,* which research the gain of performance. The fixed-size model is of concern in our following discussions.

Fixed-size Model the fixed-size model is popular since it is more applicable than the others. In Amdahl's law the amount of the computational work stays constant as the number of the processor increases. The fixed-size model can be directly derived from the asymptotic speedup model if the number of processors p is bounded and $p < k$. Then,

$$t_k(p) = \frac{W_k}{k \times \Delta} \left\lceil \frac{k}{p} \right\rceil \qquad (1-9)$$

Even in such a case $\left\lceil \frac{k}{p} \right\rceil = 1$, the equation still holds. Hence,

$$T(p) = \sum_{k=1}^{m} \frac{W}{k \times \Delta} \left\lceil \frac{k}{p} \right\rceil \qquad (1-10)$$

And the speedup is:

$$S(p) = \frac{T(1)}{T(p)} = \frac{\sum_{k=1}^{m} W_k}{\sum_{k=1}^{m} \frac{W_k}{k \times \Delta} \left\lceil \frac{k}{p} \right\rceil} \qquad (1-11)$$

If communication latency and the other system overhead are included, the speedup can be rewritten as:

$$S(p) = \frac{T(1)}{T(p)} = \frac{\sum_{k=1}^{m} W_k}{\sum_{k=1}^{m} \frac{W_k}{k \times \Delta} \left\lceil \frac{k}{p} \right\rceil + Q(p)} \qquad (1-12)$$

Where $Q(p)$ is the total overhead.

Let us simplify the model to make the workload (problem size) only contain two parts: a sequential part (DOP=1) and a perfectly parallel part (DOP=p), i.e. $W_k = 0$ for $1 < k < p$ in the parallelism profile, then the speedup can be written as:

$$S(p) = \frac{W_1 + W_p}{W_1 + W_p / p} \qquad (1-13)$$

Assuming that $W_1 + W_p = 1$ and α is $W_1 / (W_1 + W_p)$, which represents the percentage of the sequential fraction in total workload, thus, the speedup becomes:

$$S(p) = \frac{1}{\alpha + (1-\alpha)/p} = \frac{p}{1+(p-1)\times\alpha} \qquad (1-14)$$

Eq.(1-14) is well known as Amdahl's Law. The Amdahl's law [Amdahl67] is based on the common observation that every algorithm in nature has a sequential component that will limit the speedup achievable on a parallel system.

1.4 Implementation Environment (DAS-2)

Since this project focuses on implementing Grid Speedup theory, the experimental environment we use is DAS-2 Grid. DAS-2 is a real computational grid on which we can run our applications. Comparing with the traditional distributed computing environments, girds are new successors, but we should realize the fact that they are fundamentally different. Unlike the traditional distributed computing environments, a grid environment is dynamitic and unpredictable.

1.4.1 Hardware Configuration

$DAS-2$ is the second generation of DAS supercomputer, which is an acronym of **D**istributed **A**SCI **S**upercomputer designed by the **A**dvanced **S**chool for **C**omputing **I**maging (ASCI)[DAS-2web]. DAS-2 consists of 5 clusters with a total of 200 nodes, respectively located at the five Dutch universities.

Each node consists of:

·Two 1-GHz Pentium-IIIs

·At least 1 GB RAM

·At least a 20 GB local IDE disk (Integrated Disk Electronics)

·A Myrinet interface card

·A Fast Ethernet interface (on-board)

The five clusters are connected with the Dutch University Internet Backbone. The nodes within a local cluster are interconnected through a Myrinet-2000 network, which is used as the high-speed interconnection, mapped into user-space. In addition, Fast Ethernet is used as OS network (file transport). [DAS-2web].

Myrinet is based on packet-switching technology, where the packets are wormhole-routed through a network, including switching elements and network interface cards (NIC). Each NIC provides flexible programmability for designing communication sofrtware.

1.4.2 Software Configuration

The operating system the DAS-2 runs is RedHat Linux. The Globus toolkit, one of the most popular middleware packages today, has been installed on all DAS-2 clusters for the purpose that one of the main research areas for DAS-2 is Grid Computing. With such a middleware, the aim is to integrate a wide variety of machines effectively, including supercomputers, storage systems, data resources and special devices such as scientific instruments and visualization equipments.

The MPI (**M**essage **P**assing **I**nterface) implementation used in a Globus environment is MPICH-G2. There exist several mpich variants in DAS-2 environment (*Table* −1.1). Nodes in one cluster are interconnected with both Myrinet and Fast Ethernet. Cluster-to-cluster is connected with Fast Ethernet. mpich-p4-gcc and mpich-gm-gcc only can be used in one cluster. But mpich-p4-gcc uses Fast Ethernet and mpich-gm-gcc uses Myrinet. mpich-g2-ip-gcc and mich-g2-gm-gcc not only can be used in one cluster, but also can be used among clusters. However, both mpich-g2-ip-gcc and mpich-g2-gm-gcc use Myrinet in one cluster and use Fast Ethernet among clusters. In addition, mpich-gm-gcc and mpich-g2-gm-gcc are default settings on DAS-2.

We also point out that GM is a commercial open source user-level networking protocol running on the top of the Myrinet network from Myrinet. GM supports both *send/receive* and *Remote Direct Memory Access* (RDMA). The send/receive mode is a two-side operation, in which both the sender and the receiver of a message are involved in communication [Myrinetweb].

Table −1.1 MPI Variants and their supported networks

MPI Variants	Inter-Cluster		Intra-Cluster
	Fast Ethernet	Myrinet-2000	Fast Ethernet
mpich-p4-gcc	Yes	No	No
mpich-gm-gcc	No	Yes	No
mpich-g2-ip-gcc	No	Yes	Yes
mpich-g2-gm-gcc	No	Yes	Yes

1.5 Summary

The classical parallel computing metrics really do not function beyond one cluster. Hence, it is necessary to extend the traditional metrics to a real grid environment. Hoekstra did the pioneer work and in the following discussions, the focus of this project is on this grid speedup theory.

Chapter II Grid Speedup Theory

This section discusses the grid speedup theory introduced by Hoekstra [Hoekstra05]. The key idea behind the grid speedup theory is the strategy to portion the workload among C CEs (Computing Elements i.e. clusters). The hierarchical decompositions are conducted in the grid speedup theory.

2.1. Hierarchical Decompositions

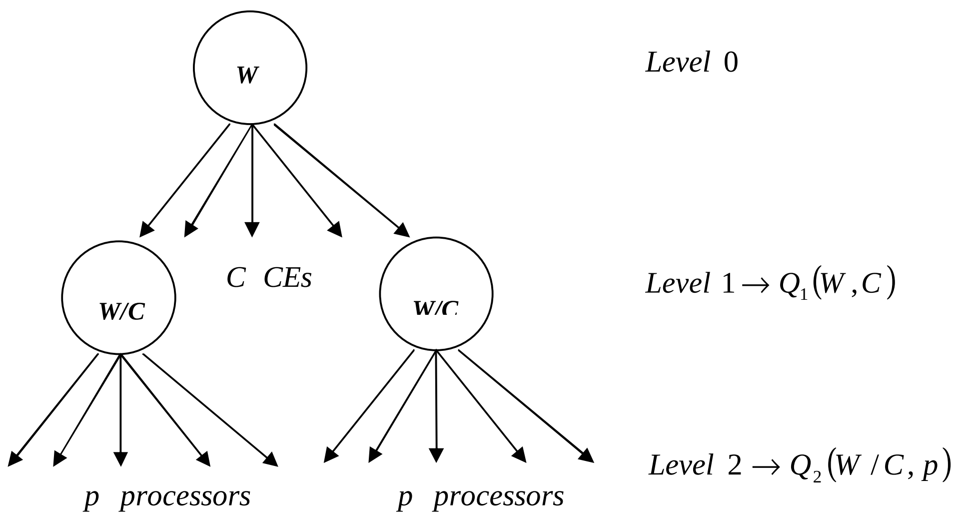


Fig.2.1 Hierarchical Decomposition Strategy

Fig.2.1 [Hoekstra05] shows that a given workload W is decomposed among C CEs and the workload portion W/C is decomposed once again among p processors inside each cluster. Finally, each processor locally has a piece of the workload $W/(C\times p)$ under the assumption of a homogeneous architecture, which means that each cluster has the same processors participating in computation and each processors has the same computing capacity Δ or τ_{calc}. We also note that two kinds of overhead $Q_1(W,C)$ and $Q_2(W/C,p)$ arise respectively from level 1 and level 2. Both of them include communication cost and load imbalance.

The feasibility of the hierarchical decompositions has to be taken into account firstly. The level 1 is a macro-strategy and conversely, the level 2 is a micro-strategy. The macro-decomposition can be done through the Globus toolkit to portion the workload W as evenly as possible among clusters. The micro-decomposition can be mapped onto the Cartesian topology defined by MPI. We will discuss the details in Chapter 4.

2.2. Scalability

According to Fig.2.1, we directly have:

$$T_{C,p}(W_p) = \frac{W_p / C}{p \times \Delta} + Q_1(W_p, C) + Q_2(W_p / C, p) \qquad (2-1)$$

Where

W_p : A given parallel workload

$T_{C,p}(W_p)$: Execution time over C clusters and p processors used in each cluster

$Q_1(W_p, C)$: Overhead from the first-level decomposition

$Q_2(W_p / C, p)$: Overhead from the second-level decomposition

C: Number of clusters

Δ : Computing capacity

p : Number of processors

On a single processor within one cluster, we have:

$$T_{1,1}(W_p) = \frac{W_p}{\Delta} \qquad (2-2)$$

Where

$T_{1,1}(W_p)$: Execution time on a single processor within one cluster

Then we obtain the speedup:

$$S_p^C = \frac{T_{1,1}(W_p)}{T_{C,p}(W_p)} = \frac{\frac{W_p}{\Delta}}{\frac{W_p}{p \times \Delta \times C} + Q_1(W_p, C) + Q_2(W_p / C, p)}$$

$$= \frac{pC}{1 + \frac{pC\Delta}{W_p} Q_1(W_p, C) + \frac{pC\Delta}{W_p} Q_2(W_p / C, p)}$$

$$= \frac{pC}{1 + \frac{Q_1(W_p, C)}{\frac{W_p}{pC\Delta}} + \frac{Q_2(W_p / C, p)}{\frac{W_p}{pC\Delta}}}$$

$$= \frac{pC}{1 + \frac{Q_1(W_p, C) + Q_2(W_p / C, p)}{\tau(W_p, p, C)}} \qquad (2-3)$$

Where

S_p^C: Relative speedup

$$\tau(W_P, p, C) = \frac{W_P}{pC\Delta}.$$

Note that superscript *C* means the decomposition over *C* clusters.

The denominator term $pC\Delta$ indicates the total computing capacity involved and the numerator W denotes the total workload. Consequently, $\tau(W, p, C)$ has the time unit (e.g. in second), meaning the execution time consuming on the pure computational workload. In Eq.(2-3), the ratio of $[Q_1(W,C)+Q_2(W/C,p)]$ to $\tau(W, p, C)$ is dimensionless. If $C=1$ (namely, one cluster), then $Q_1(W,C)=0$, again reducing to the general parallel computing. We can predict that the overhead term $Q_1(W,C)$ among clusters would extremely effect on the gain of performance, comparing to the overhead term $Q_2(W/C,p)$ mostly originating from local communication inside one cluster under the assumption of the performance of the present networks.

We also have noted that Eq.(2-3) does not embody the relative values of the execution time among *C* clusters to the execution time within one cluster.

2.3. Grid Speedup

The reasons why people turn to the parallel computing are mentioned by [Hoekstra05]:

·bounded computing capacity, meaning that a single processor will lead to a bounded computing capacity.

·bounded memory space, meaning that a single processor has a limited memory space which does not fit the memory consumption of a huge dataset produced by an application.

Also based on the similar reasons, we decompose the workload over more than one **CE**. The concept of Grid Speedup originates from this decomposition strategy. Grid Speedup and Grid Efficiency are defined as:

$$G_p^C = \frac{T_{1,p}(W_p)}{T_{C,p}(W_p)} \qquad (2-4)$$

$$\gamma_p^C = \frac{G_p^C}{C} = \frac{1}{C}\frac{T_{1,p}(W_p)}{T_{C,p}(W_p)} \qquad (2-5)$$

Where

G_p^C : Grid speedup

γ_p^C : Grid efficiency

W_p : A given parallel workload

$T_{1,p}(W_p)$: Execution time on p processors inside one cluster

$T_{C,p}(W_p)$: Execution time over C clusters and p processors used in each cluster

Substitute Eq. (2-4) with Eq.(2-1) and then we have:

$$G_p^C = \frac{T_{1,p}(W_p)}{T_{C,p}(W_p)} = \frac{\frac{W_p}{p\Delta} + Q_2(W_p, p)}{\frac{W_p}{pC\Delta} + Q_1(W_p, C) + Q_2(W_p / C, p)}$$

$$= \frac{C \times \left[\frac{W_p}{p\Delta} + Q_2(W_p, p)\right]}{\left[\frac{W_p}{p\Delta} + Q_2(W_p, p)\right] + C \times Q_1(W_p, C) + C \times [Q_2(W_p / C, p) - Q_2(W_p, p)]}$$

$$= \frac{C}{1 + g_1(W_p, p, C) + g_2(W_p, p, C)} \qquad (2\text{-}6)$$

Where

$$g_1(W_p, p, C) = C \times \frac{Q_1(W_p, C)}{T_{1,p}(W_p)} \qquad (2-7)$$

$$g_2(W_p, p, C) = \frac{C \times Q_2(W_p / C, p) - Q_2(W_p, p)}{T_{1,p}(W_p)} \qquad (2-8)$$

Hoekstra argued that in Eq.(2-6) the term $g_2(W_p, p, C)$ is really meaningful since it can mathematically have three possibilities: zero, positive or negative. Here let us review them once again.

Case 1. $C \times Q_2(W_P / C, p) = Q_2(W_p, p)$: This indicates that the overhead from C clusters is scalable with a factor of $1/C$ to the overhead from one cluster. In this case, we have $g_2(W_P, p, C) = 0$ and the grid speedup can be rewritten as:

$$G_p^C = \frac{C}{1 + g_1(W_P, p, C)} \qquad (2-9)$$

Case 2. $C \times Q_2(W_P / C, p) > Q_2(W_p, p)$: This suggests that the term $g_2(W_P, p, C)$ negatively affects the grid speedup. We can imagine in this situation the overhead within one cluster seems to be hidden and even as if it would not exist for the reason that the overhead among clusters is much larger.

Case 3. $C \times Q_2(W_P / C, p) < Q_2(W_p, p)$: This implies that the term $g_2(W_P, p, C)$ has a positive effect on the grid speedup. We can predict that it theoretically leads to super linear grid speedups since decomposing the same workload among C clusters causes less overhead than doing among p processors in one cluster. It is far more likely to happen if network communication is as ideal as possible.

2.4. Case Studies

The previous discussions are implicit and from now on we will setup an analytical model that we can explicitly discuss. In this subsection, the model we use is to solve the Laplace equation with the Dirichlet boundary condition in 2D square domain

Without loss of generality, we assume that Eq.(2-1) has $(0 \le x, y \le 1)$.

$$\nabla^2 u = (\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = 0 \qquad (2\text{-}10)$$

We assume that Eq.(2-1) has the same meaning as in physics, namely, the site of y=1 is source , y=0 is sink and u is the temperature values of a point in domain. In addition, we also assume that Eq.(2-1) has a periodic boundary condition in the x direction. We further assume that the square domain is discretized into $n \times n$ grid points by using FDM (**F**inite **D**ifference **M**ethod), where n is the number of grid points along x or y direction.

2.4.1 Case I

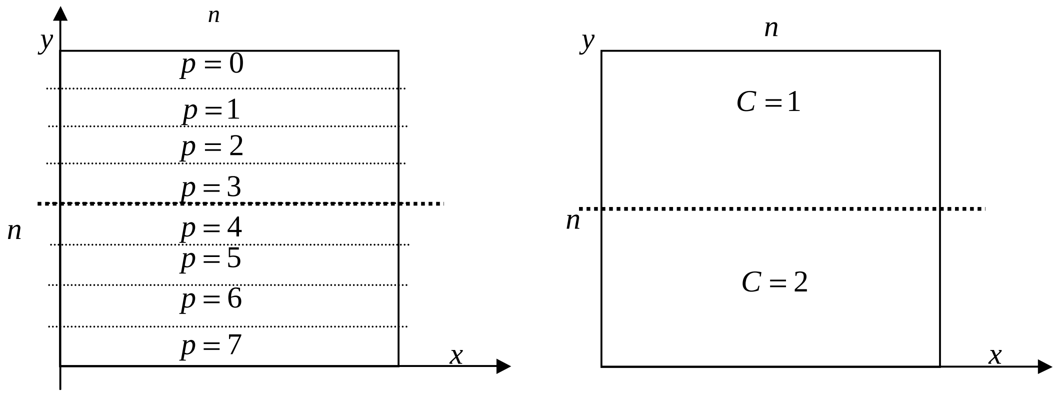


Fig.2.2 Case I: $C = 2$

Left : Level 2 *Right : Level* 1

In Case I, both Level 1 and Level are decomposed in row-wise. Fig.2.2 and Fig.2.3 show 2 possible sub-cases: $C = 2$ *and* $C \geq 3$.

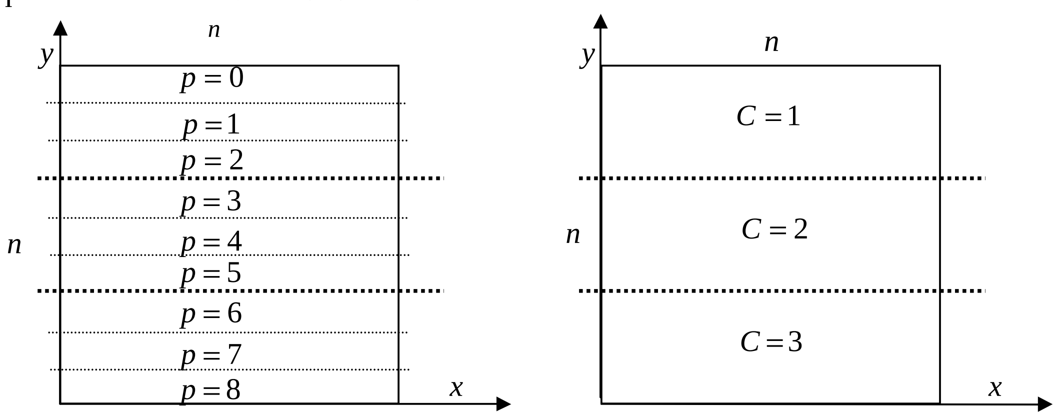


Fig.2.3 Case I: $C = 3$

Left : Level 2 *Right : Level* 1

For convenience, we group 2 sub-cases into 2 situations: one is for $C = 2$, Fig.2.2 and another is for $C \geq 3$, Fig.2.3. In each group, several specific cases are classified according to the number of processors in each cluster. In the following discussions, we consecutively investigate them.

Group 1. $C = 2$

In Fig.2.2, the right shows that the workload is decomposed between 2 clusters and the left illustrates that each portion of the workload is decomposed once again in the same way as the right. Consequentially, there exist 3 possibilities according to the number of processors involved in each cluster.

1. $p = 1$

The execution time on one cluster with 1 processor can be calculated by:

$$T_{1,1} = \frac{n^2}{\Delta}$$

On two clusters:

$$T_{2,1} = \frac{n^2}{2\Delta} + T_{comm}^{grid}$$

Grid speedup:

$$G_1^2 = \frac{T_{1,1}}{T_{2,1}} = \frac{\frac{n^2}{\Delta}}{\frac{n^2}{2\Delta} + T_{comm}^{grid}} = \frac{2}{1 + \frac{2T_{comm}^{grid}}{\frac{n^2}{\Delta}}} \qquad (2-11)$$

In such a situation, communication only occurs between 2 clusters. Totally 2 n grid points will exchange globally.

2. $p = 2$

On one cluster with 2 processors:

$$T_{1,2} = \frac{n^2}{2 \times \Delta} + T_{comm}^{cluster}$$

$(T_{comm}^{cluster} \propto 2n)$

On two clusters with 2 processors in each cluster:

$$T_{2,2} = \frac{n^2}{2 \times 2 \times \Delta} + T_{comm}^{grid} + T_{comm}^{cluster}$$

$(T_{comm}^{grid} \propto 2n)$

Grid speedup:

$$G_2^2 = \frac{T_{1,1}}{T_{2,2}} = \frac{\frac{n^2}{2\times\Delta} + T_{comm}^{cluster}}{\frac{n^2}{2\times 2\times\Delta} + T_{comm}^{grid} + T_{comm}^{cluster}}$$
$$= \frac{2\left(\frac{n^2}{\Delta} + 2T_{comm}^{cluster}\right)}{\frac{n^2}{\Delta} + 2T_{comm}^{cluster} + 2T_{comm}^{cluster} + 4T_{comm}^{grid}} = \frac{2}{1 + \frac{2T_{comm}^{cluster} + 4T_{comm}^{grid}}{\frac{n^2}{\Delta} + 2T_{comm}^{cluster}}} \qquad (2-12)$$

Which is rewritten as

$$G_2^2 = \frac{2}{1 + \frac{(2+4\alpha)}{\beta + 2}}$$

with

$$\alpha = \frac{T_{comm}^{grid}}{T_{comm}^{cluster}}$$

$$\beta = \frac{n^2}{\Delta T_{comm}^{cluster}}$$

Where $T_{comm}^{cluster}$ is overhead, which is a combination of the communication latency and the data-exchange time in one CE. T_{comm}^{grid} is the communication time between two neighboring clusters.

3. $p \geq 3$

On one cluster with not less than 3 processors in each cluster:

$$T_{1,\,p} = \frac{n^2}{p \times \Delta} + 2T_{comm}^{cluster}$$

On two clusters with not less than 3 processors in each cluster:

$$T_{2,\,p} = \frac{n^2}{p \times 2 \times \Delta} + T_{comm}^{grid} + T_{comm}^{cluster}$$

Grid speedup:

$$G_p^2 = \frac{T_{1,\,p}}{T_{2,\,p}} = \frac{\frac{n^2}{p\times\Delta} + 2T_{comm}^{cluster}}{\frac{n^2}{p \times 2 \times \Delta} + T_{comm}^{grid} + T_{comm}^{cluster}}$$

$$= \frac{2\left(\frac{n^2}{\Delta} + 2pT_{comm}^{cluster}\right)}{\frac{n^2}{\Delta} + 2pT_{comm}^{cluster} + 2pT_{comm}^{grid}} = \frac{2}{1 + \frac{2\alpha}{\beta + 2}} \qquad (2-13)$$

$$\alpha = \frac{T_{comm}^{grid}}{T_{comm}^{cluster}}$$

$$\beta = \frac{n^2}{p\Delta T_{comm}^{cluster}}$$

Group 2. $C \geq 3$

Fig.2.3 shows that the workload is decomposed in row-wise way among 3 clusters.

1. $p=1$

The execution time on one cluster with 1 processor can be calculated by:

$$T_{1,1}=\frac{n^2}{\Delta}$$

On three clusters:

$$T_{C,1}=\frac{n^2}{C\Delta}+2T_{comm}^{grid}$$

$$(T_{comm}^{grid}\propto 2n)$$

$$G_1^C=\frac{T_{1,1}}{T_{C,1}}=\frac{\frac{n^2}{\Delta}}{\frac{n^2}{C\Delta}+2T_{comm}^{grid}}=\frac{C}{1+\frac{2CT_{comm}^{grid}}{\frac{n^2}{\Delta}}} \qquad (2-14)$$

2. $p=2$

On one cluster with 2 processors:

$$T_{1,2}=\frac{n^2}{2\times\Delta}+T_{comm}^{cluster}$$

On three clusters with 2 processors in each cluster:

$$T_{C,2}=\frac{n^2}{C\times 2\times\Delta}+T_{comm}^{grid}+T_{comm}^{cluster}$$

$$G_2^C=\frac{T_{1,2}}{T_{C,2}}=\frac{\frac{n^2}{2\times\Delta}+T_{comm}^{cluster}}{\frac{n^2}{C\times 2\times\Delta}+T_{comm}^{grid}+T_{comm}^{cluster}}$$

$$=\frac{C\left(\frac{n^2}{2\times\Delta}+T_{comm}^{cluster}\right)}{\frac{n^2}{2\times\Delta}+T_{comm}^{cluster}+(C-1)T_{comm}^{cluster}+C\times T_{comm}^{grid}}=\frac{C}{1+\frac{C(1+\alpha)-1}{\beta+1}} \qquad (2-15)$$

$$\alpha=\frac{T_{comm}^{grid}}{T_{comm}^{cluster}}$$

$$\beta=\frac{n^2}{\Delta T_{comm}^{cluster}}$$

3. $p \geq 3$

On one cluster with not less than 3 processors in each cluster:

$$T_{1,p} = \frac{n^2}{p \times \Delta} + 2T_{comm}^{cluster}$$

On two clusters with not less than 3 processors in each cluster:

$$T_{C,p} = \frac{n^2}{p \times C \times \Delta} + T_{comm}^{grid} + T_{comm}^{cluster}$$

Grid speedup:

$$G_p^C = \frac{T_{1,p}}{T_{C,p}} = \frac{\frac{n^2}{p \times \Delta} + 2T_{comm}^{cluster}}{\frac{n^2}{p \times C \times \Delta} + T_{comm}^{grid} + T_{comm}^{cluster}}$$

$$= \frac{C\left(\frac{n^2}{p \times \Delta} + 2T_{comm}^{cluster}\right)}{\frac{n^2}{p \times \Delta} + 2T_{comm}^{cluster} + (C-2)T_{comm}^{cluster} + CT_{comm}^{grid}} \qquad (2-16)$$

$$= \frac{C}{1 + \frac{C(1+\alpha) - 2}{\beta + 2}}$$

$$\alpha = \frac{T_{comm}^{grid}}{T_{comm}^{cluster}}$$

$$\beta = \frac{n^2}{p \Delta T_{comm}^{cluster}}$$

Comparing Eq.(2-16) with Eq.(2-1), we have the overhead functions:

$$Q_1(W, C) = 2T_{comm}^{grid} \qquad (2-17)$$

$$Q_2(W / C, p) = T_{comm}^{cluster} \qquad (2-18)$$

We observe that this is the *Case 1* in the previous discussions. Hence, we obtain the grid speedup as follows:

$$G_p^C = \frac{C}{1+g_1(W_P, p, C)} = \frac{C}{1+\frac{CT_{comm}^{grid}}{\frac{n^2}{p\Delta}+2T_{comm}^{cluster}}}$$

$$= \frac{C}{1+\frac{C\alpha}{\beta+2}} \qquad (2\text{-}19)$$

Where

$$\alpha = \frac{T_{comm}^{grid}}{T_{comm}^{clsuter}} \qquad (2-20)$$

$$\beta = \frac{n^2}{p\Delta T_{comm}^{clsuter}} \qquad (2-21)$$

Note that two dimensionless parameters α and β are also dependent on the parameter n since neither T_{comm}^{grid} nor $T_{comm}^{clsuter}$ is independent of the parameter n. Here, for simplicity, we just neglect it, but we must take it into consideration in a real implementation. In our following experiments, both terms T_{comm}^{grid} and $T_{comm}^{clsuter}$ are based on Hockney model [Hockney94].This implies these two terms are extremely sensitive to message length (i.e. problem size) and also hardware parameters. [Hockney94].

The parameter α means the imbalance in the hierarchical communication between inter- and intra CE communication. The parameter β is more complicated because it contains 2 variables. Therefore, we only can confirm that the larger the value of the parameter β is, the bigger the computing capacity of the cluster is.

2.4.2 Case II

Likewise, this situation can be grouped as 2 cases: one is for 2 clusters and another is for more than 2 clusters. In Fig.2.3, theoretically, this case is the same as the row-wise decompositions for level 1 and level 2. But practically, there exists some difference in their implementations. Even though 2n grid points will be exchanged, they reside in distributed positions in memory (Assume C=2). In order to exchange, these grid points have to be packed firstly and unpacked after each processor receive them. We will leave them for the later experiments.

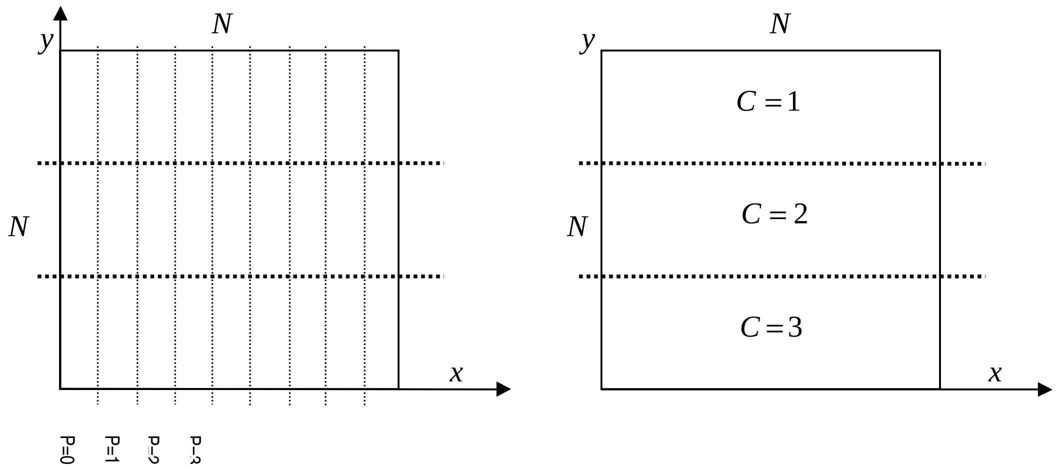


*Fig.*2.4 *Case II:C* = 3

Left: Level 2 *Right Level*1

2.4.3 Case III

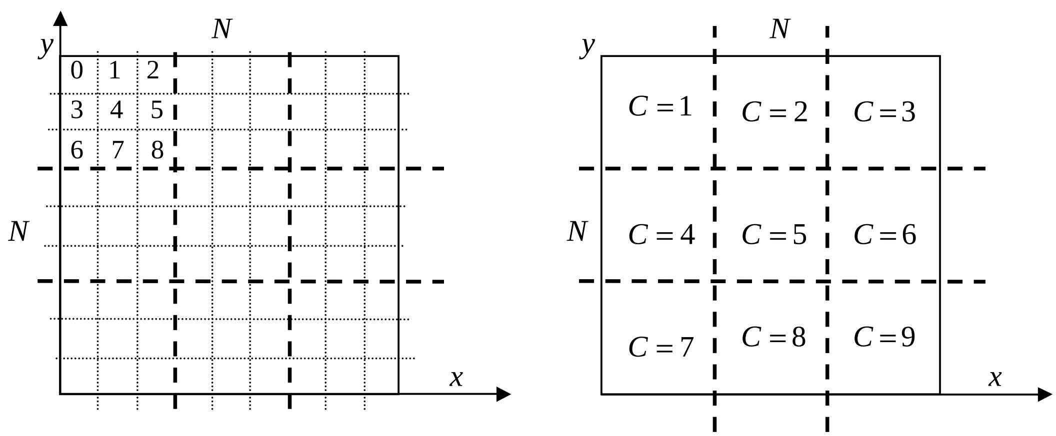


*Fig.*2.5 *Case III: C* = 9

Left: Level 2 *Right Level*1

As shown in Fig.2.4, both level 1 and level 2 are decomposed in 2D. Note that the number of the cluster we may take should be a perfect square integer (9 clusters at least) and the number of the processors inside one *CE* also should be a perfect square integer like 1, 4 and 9.

The execution time on one CE can be calculated by:

$$T_{1,\,p} = \frac{n^2}{p \times \Delta} + 4T_{comm}^{cluster} \qquad (2-22)$$

$$(T_{comm}^{cluster} \propto \frac{2n}{\sqrt{p}})$$

Note that the factor4 means 4 communications for exchanging data with its 4 neighbors if the number of CEs or the processors involved in computing is more than 9.

And on more than one CE, we have:

$$T_{C,p} = \frac{n^2}{pC\Delta} + 2T_{comm}^{grid} + 2T_{comm}^{cluster^{\sim}} \qquad (2-23)$$

$$(T_{comm}^{grid} \propto \frac{2n}{\sqrt{p \times C}})$$

$$(T_{comm}^{cluster^{\sim}} \propto \frac{2n}{\sqrt{p \times C}})$$

And with the overhead functions:

$$Q_1(W, C) = 2T_{comm}^{grid} \qquad (2-24)$$

$$Q_2(W / C, p) = 2T_{comm}^{cluster^{\sim}} \qquad (2-25)$$

Consequently, we get the grid speedup as follows:

$$G_p^C = \frac{T_{1,p}}{T_{C,p}} = \frac{\frac{n^2}{p \times \Delta} + 4T_{comm}^{clsuter}}{\frac{n^2}{pC\Delta} + 2T_{comm}^{grid} + 2T_{comm}^{clsuter^{\sim}}}$$

$$= \frac{C\left(\frac{n^2}{p \times \Delta} + 4T_{comm}^{clsuter}\right)}{(\frac{n^2}{p \times \Delta} + 4T_{comm}^{clsuter}) + 2CT_{comm}^{grid} + (2CT_{comm}^{clsuter^{\sim}} - 4T_{comm}^{clsuter})}$$

$$= \frac{C}{1 + \frac{2CT_{comm}^{grid} + 2(CT_{comm}^{clsuter^{\sim}} - T_{comm}^{clsuter})}{\frac{n^2}{p \times \Delta} + 4T_{comm}^{clsuter}}} = \frac{C}{1 + \left[\frac{2C\alpha + 2(C\xi - 2)}{\beta + 4}\right]} \qquad (2-26)$$

Where

$$\xi = \frac{T_{comm}^{cluster^{\sim}}}{T_{comm}^{clsuter}}$$

Note that the parameters α and β have the same definitions as the previous subsection. But they contain different parameters in comparison with Eq.(2-20)and

Eq.(2-21). In Eq.(2-26), α is of dependence on C and p as well as problem size n and β is dependent both on problem size n and p.

We investigate one special situation. We assume that there are 9 clusters and there is 1 processor available in each cluster. We immediately find that all communication takes place among clusters and communication cost will be extremely expensive since all is global communication ($Q_2 = 0$). In such a case, we obtain the grid speedup:

$$G_1^C = \frac{T_{1,1}}{T_{C,1}} = \frac{\frac{n^2}{\Delta}}{\frac{n^2}{C\Delta} + 4T_{comm}^{grid}} = \frac{C\left(\frac{n^2}{\Delta}\right)}{\frac{n^2}{\Delta} + 4CT_{comm}^{grid}}$$

$$= \frac{C}{1 + \frac{4CT_{comm}^{grid}}{\frac{n^2}{\Delta}}} \qquad (2-27)$$

In Eq.(2-27), $\frac{8n}{\sqrt{C}}$ grid points will be exchanged at most. In fact, only 3 clusters may exchange $\frac{8n}{\sqrt{C}}$ points with its 4 neighbors and the others will only exchange $\frac{6n}{\sqrt{C}}$ with its 3 neighbors. Eq.(2-27) shows an extreme situation and in practice nobody would think it could be cheap at the price.

2.5 Summary

This section reviewed the Grid Speedup Theory introduced by [Hoekstra05].The theory itself is an innovation and also an extension on the *classical* speedup theory. We have realized that how many processors involved in cluster-to-cluster communication is a very important factor to the gain of performance in a grid environment. Moreover, not all the processors take the same communication time in running the same piece of workload and so we actually use “ceiling function” to choose the largest values. In Chapter 4, we will experiment with the cases mentioned in this section and try to validate and verify them.

Chapter III   Basic Measurements

This section presents a series of experiments for evaluating the communication performance and computing capacity of the DAS-2 (see Chapter 1 for an introduction to the DAS-2 system). All the performance results are obtained by using Pentium-based clusters running the Linux operating systems.

3.1 Communication Levels

As illustrated in Fig.3.1, there exist 5 communication levels on a Grid:

· *Level* 1 : intra-processor communication (CPU ←→Cache←→Memory)

· *Level* 2 : inter-processor communication within 1node.

· *Level* 3 : inter-node communication within 2 node

· *Level* 4 : inter-cluster communication

· *Level* 5 : inter-local-grid communication

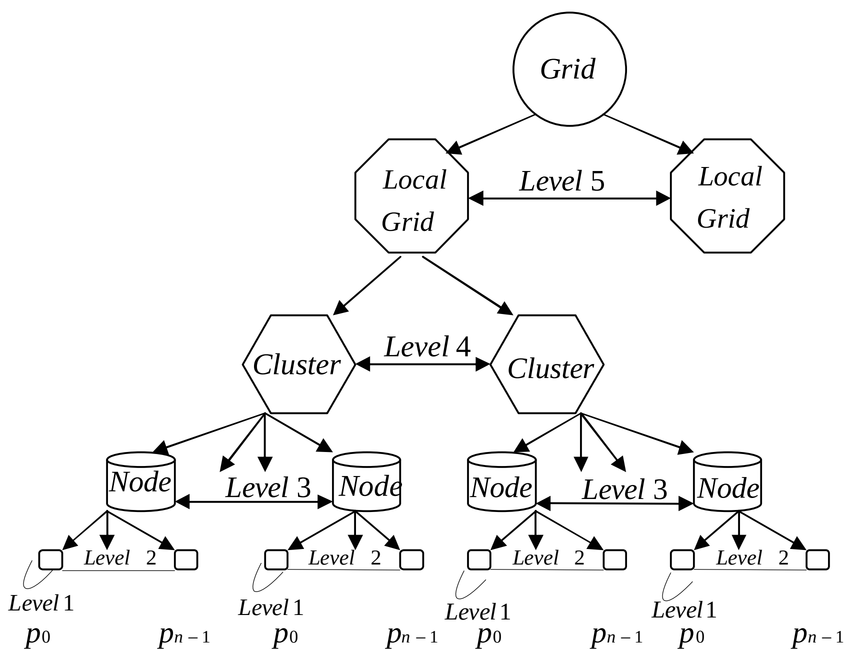


Fig.3.1 Communication Levels on a Grid

A local Grid, like the DAS-2 as mentioned in Chapter 1, usually consists of several clusters. TheDAS-2 contains 5 clusters, which are located in 5 Dutch universities. Each cluster has a homogeneous architecture, but the number of the nodes in each cluster varies. In fact, Grid Computing can be seen as the evolution of local Grid Computing to the global scale, made possible by the advent of very high-speed connection infrastructure, and of powerful processors.

For $Level\,1$, the communication cost contains the cache misses, CPU stalls and other factors from executable applications. The parameter corresponding to $Level\,1$ is τ_{calc}, which is defined as *computing capacity*. In the following discussions, we will give an insight into the parameter τ_{calc}.

For $Level\,2$ and $Level\,3$, the communication cost is theoretically a bit different. The two processors involved in $Level\,2$ are equipped on the same motherboard, but $Level\,3$ are not. For simplicity, we assume that the communication costs of $Level\,2$ and $Level\,3$ are the same. We will see that our assumption is reasonable in the following subsection.

For $Level\,4$, the communication consumption will be dominant, comparing to the sum of the communication cost from $Level\,1$, $Level\,2$ and $Level\,3$. All communication levels are regarded as point-point communication mode.

For $Level\,5$, it is beyond our discussions in this project.

3.2 Experiment Setup

In this subsection, two experiments are designed to measure the hardware parameters. One is a pingpong-like model used to measure the communication cost and bandwidth of the point-to-point communication both within one cluster and between 2 clusters. Another is to estimate the computing capacity by solving the Lapalce Equation with the Dirichlet boundary condition in 2D square domain.

3.2.1 Point-to-Point Communication

Many parallel communication models have been frequently used to predict performance such as Hockney model [Hockney94], LogP [Culler93], LogGP [Alexandrov95]. We choose Hockney model as our point-to-point communication model , defined as:

$$T_{comm} = \alpha + \beta \times L \qquad (3-1)$$

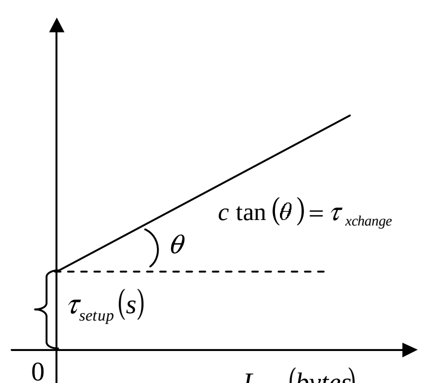


Fig.3.2 Point-to-point Communication Model

Where α is the setup time for communication and β is the transfer time per byte; L is the message length in bytes.

That means that one processor acts as a sender to echo a message of L and another processor acts as a receiver to receive the message. Then they do a flip-flop: the receiver acts as a sender to echo back the same message. Hence we rewrite Eq.(3-1) as:

$$T_{comm}(L) = \tau_{send} + \tau_{receive} + 2 \times L \times \tau_{exchange} \qquad (3-2)$$

Where

τ_{send} : The setup communication latency from the sender.

$\tau_{receive}$: The setup communication latency from the receiver.

We further assume that τ_{send} is equal to $\tau_{receive}$. Therefore, we have:

$$T_{comm}(L) = \tau_{setup} + 2L \times \tau_{exchange} \qquad (3-3)$$

Where

$$\tau_{setup} = (\tau_{send} + \tau_{receive})$$

Under the assumption of $Level\ 2 = Level\ 3$, according to Eq.(3-3), let us define the network parameters for the communication levels $Level\ 3$ and $Level\ 4$ described in Fig.3.1. These network parameters should be constants in theory because they are related to the hardware. But in practice, they could not keep constant because of the software latency.

$\tau_{setup}^{cluster}$: Average setup communication latency in one cluster

$\tau_{exchange}^{cluster}$: Average transferring cost per byte in one cluster

τ_{setup}^{grid} : Average setup communication latency between two clusters

$\tau_{exchange}^{grid}$: Average transferring cost per byte between two clusters

In subsection 3.3.1, we will experiment with Eq.(3-3) to measure the parameters we defined above.

3.2.2 Computing Capacity

We can obtain the computing capacity parameter τ_{calc} by solving the Laplace Equation with an iterative scheme. The reason for using iterative method is based on the consideration that under a given convergent condition we can exactly know how many iterations our program may have on a single processor.

The key idea behind the model is that the whole square domain is discretized into $n \times n$ grid points with FDM (**F**inite **D**ifference **M**ethod)., where n is the grid points along each direction (x or y). In each iteration, $n \times n$ grid points are updated consecutively. Hence, τ_{calc} can be defined as:

$$\tau_{calc} = \frac{T_{1,1}(W_p)}{n^2} \quad Or \; T_{1,1}(W_p) = n^2 \tau_{\text{calc}} \qquad (3-4)$$

Where $T_{1,1}(W_p)$ is the execution time to sweep $n \times n$ grid points in one iteration. Obviously, τ_{calc} is the cost used to update one grid point and it is a constant if our algorithm in our parallel program has been determined.

3.3 Empirical Results

In this subsection, we did the experiments to obtain the parameters we defined in the previous subsections.

3.3.1 Parameters of Network

We used a pingpong-like program to measure the network parameters including $\tau_{setup}^{cluster}$, $\tau_{exchange}^{cluster}$, τ_{setup}^{grid} and $\tau_{exchange}^{grid}$, which were defined in subsection 3.2.1. The experimental results are listed in $Table-3.1$, Fig.3.3 and Fig.3.4. We used mpich-gm in one cluster and mpich-g2-gm between two clusters. An alternative to cross clusters is mpich-g2-ip, which performs similarly as mpich-g2-gm does. In addition, we adopted the MPI synchronous communication mode, which is considered as a safe communication mode, compared with the other 3 modes. Note that the implementation within one cluster can be done in traditional manner, but between two clusters has to be done in combination with the Globus toolkit. Because of the dynamitic characteristic of network, $Table-3.1$statistically shows the average values and also includes the standard devaition. The value of each parameter sampled 20 times on each cluster and averaged the empirical data. The values of $\tau_{setup}^{cluster}$ and $\tau_{exchange}^{cluster}$ show the performance of Myrinet within one cluster, but τ_{setup}^{grid} and $\tau_{exchange}^{grid}$ demonstrate the performance of Fast Ethernet among clusters. For two processors within one cluster, they may be affiliated to the same node or not. However, the experiments show there is not much to choose between these two cases.

Fig.3.5 presents the comparisons of 4 MPI variants: mpich-gm and mpich-g2-gm are beyond all doubt the best choices. All parameters are obtained by linear regression: the coefficients are exchange time per byte and the intercepts are setup time.

$Table-3.1$

Parameter	$\tau_{setup}^{cluster}$	$\tau_{exchange}^{cluster}$	τ_{setup}^{grid}	$\tau_{exchange}^{grid}$
Average Value	8.0 ± 1.0 us	5.0 ± 2.0 ns/byte	1.6 ± 0.2 ms	90.0 ± 5.0 ns/byte

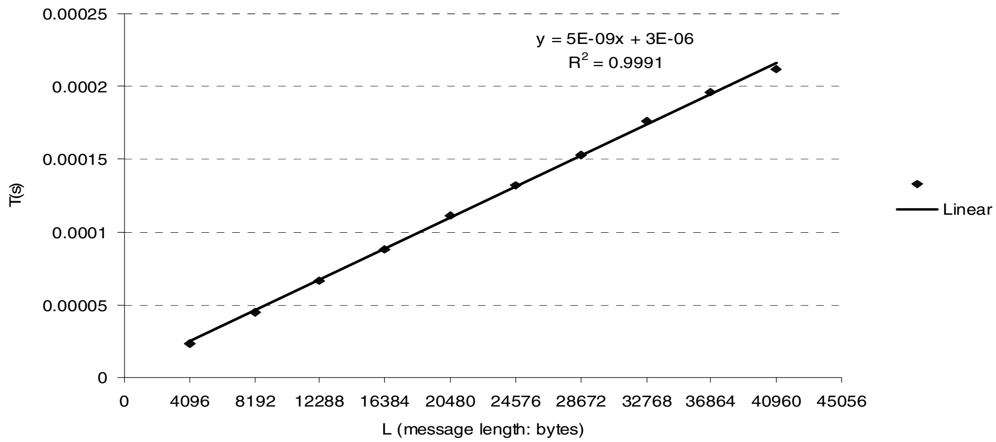


Fig.3.3 Transferring Time as a function of Message Length (fs0)

In Fig.3.3, the empirical data goes along with a line: y= 5E-09x+3E-06. The coefficient 5E-09 means 5E-09 s per byte and the constant 3E-06 suggests the setup time of 3 us for preparing to communicate. During our experiments, we noted that the violations of communication between 2 clusters probably take place since some instantaneous noises exist.

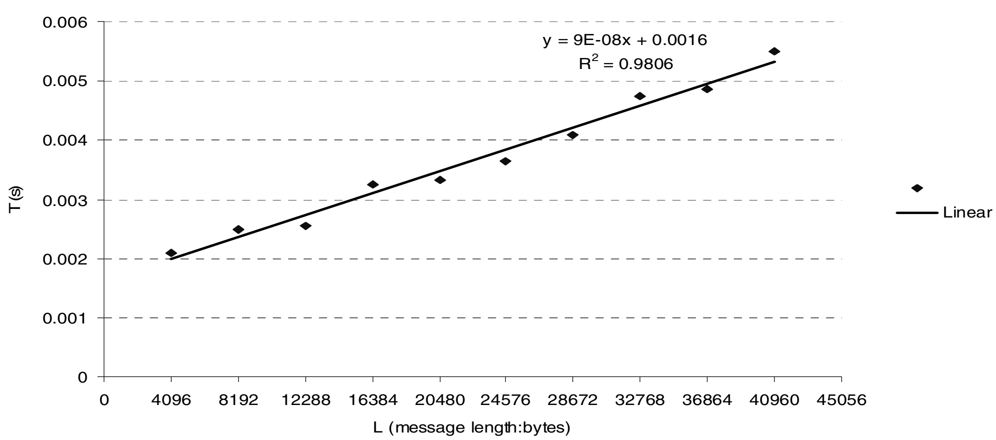


Fig.3.4 Transferring Time as a function of Message Length (fs0-fs2)

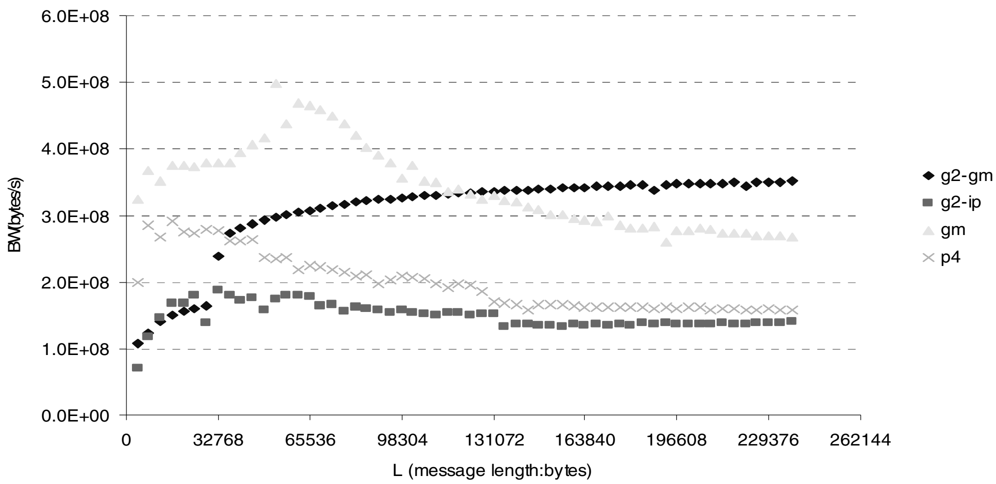


Fig.3.5 Comparisons of throughput of 4 MPI variants available on DAS-2

3.3.2 Parameter of Computing Capacity

Here, we implemented the computing capacity model and obtained the parameter of computing capacity τ_{cacl}. As explained in subsection 3.2.2, theoretically, τ_{cacl} is a constant, however as indicated in Fig.3.6, τ_{cacl} is not a fixed value, but dependent on problem size n. There exists one main reason: as problem size is small, the data will fit in the cache. It implies that τ_{cacl} is also dependent on problem sizes. $Table-3.2$ presents the statistical results over different problem sizes. We sampled more than 20 times for each problem size and averaged them. As shown in Fig.3.6, error bars are used to indicate possible fluctuation of τ_{cacl}.

Hence, we have:

$$\tau_{cacl}(n) = \begin{cases} 70 \pm 1.6 \ (ns) \ \ if\, n < 128 \\ 102 \pm 1.6 \ (ns) \ \ if\, n = 128 \\ 148 \pm 1.6 \ (ns) \ if\, n > 256 \end{cases}$$

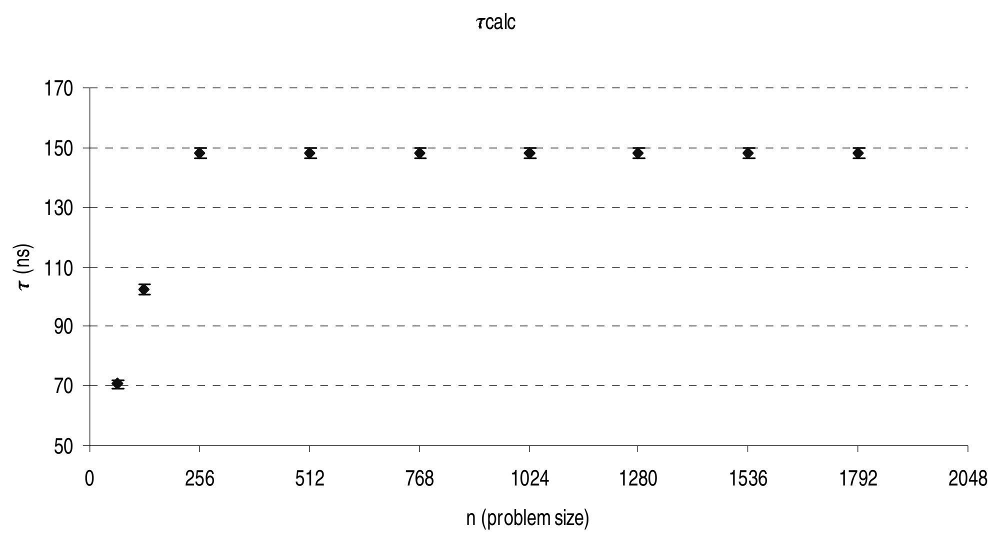


Fig.3.6 τ_{calc} *as a function of n (problem size)*

Table – 3.2 τ_{calc} *as a function of n (problem size)*

n	64	128	256	512	768	1024	1280	1536
τ_{calc}(ns)	70.5	102.3	147.3	146.8	150.6	142.0	150.3	150.2

3.4 Summary

In this section, we did a series of basic experiments based on the models we designed. And we obtained the hardware parameters such as $\tau_{setup}^{cluster}, \tau_{exchange}^{cluster}, \tau_{setup}^{grid}$, $\tau_{exchange}^{grid}$ and τ_{cacl} . In addition, we discussed some factors which may affect on the hardware parameters.

Chapter IV Measurements of Grid Speedup

The focus of this section is on measurements of Grid speedup. One model is designed to implement the grid speedup theory, which is an extension on the prototypic model we used in the previous section. Experimental results are compared to the theoretical ones obtained from the analytical models.

4.1. Model designs

In this subsection one analytical model is designed to experiment the cases we discussed in Chapter 2.

4.1.1 An Analytical Model In a 2D Square Domain

We will design an analytical model of the time complexity for solving the Time Independent Laplace Equation (2-1), which has been mentioned in subsection 2.4.

Eq.(2-1) can be discretized with FDM (**F**inite **D**ifference **M**ethod). In order to approximate the second derivatives in the spatial domain, we once again use Taylor's expansion.

$$
\begin{aligned}
u(x+\delta x, y, t) &= u(x, y, t) + \delta x \frac{\partial u(x, y, t)}{\partial x} \\
&\quad + \frac{1}{2}(\delta x)^2 \frac{\partial^2 u(x, y, t)}{\partial x^2} + O((\delta x)^3) \\
u(x-\delta x, y, t) &= u(x, y, t) - \delta x \frac{\partial u(x, y, t)}{\partial x} \\
&\quad + \frac{1}{2}(\delta x)^2 \frac{\partial^2 u(x, y, t)}{\partial x^2} + O((\delta x)^3)
\end{aligned}
\qquad (4\text{-}1)
$$

Similarly the approximation to the y direction is found. After several arithmetic conversions to both sides of Eq. (4-1), we get

$$
\begin{aligned}
\frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\delta x)^2} \\
\frac{\partial^2 u_{i,j}}{\partial y^2} &= \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(\delta y)^2}
\end{aligned}
\qquad (4\text{-}2)
$$

The spatial domain is described by the grid coordinates (i, j). Note that we neglected the small errors from Eq.(4-1) to Eq.(4-3).

Substituting Eq.(2-1) with these terms (4-3) gives us

$$0 = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\delta x)^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(\delta y)^2} \qquad (4\text{-}3)$$

Assuming that δx is equal to δy when they are small enough, we have:

$$u_{i,j} = \frac{1}{4}\left[u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}\right] \qquad (4\text{-}4)$$

Eq.(4-5) immediately suggests the Jacobi iteration

$$u_{i,j}^{(m+1)} = \frac{1}{4}\left[u_{i+1,j}^{(m)} + u_{i-1,j}^{(m)} + u_{i,j+1}^{(m)} + u_{i,j-1}^{(m)}\right] \qquad (4\text{-}5)$$

m denotes iteration indices. Because of the inherent data locality in the stencil operation, Eq.(4-5) is very suitable for parallel computing. Based on the observations on Eq.(4-5), we can conclude that each term of Eq.(4-5) is associated with its 4 neighbors in spatial domain. As depicted in Fig.4.1, the value of (i, j) is of dependence upon the values of $(i-1, j)$, $(i+1, j)$, $(i, j-1)$ and $(i, j+1)$. Namely, to update a point (i, j) we need information from its nearest neighbors, i.e. from $(i-1, j)$, $(i+1, j)$, $(i, j-1)$ and $(i, j+1)$, which results in a 5-point stencil.

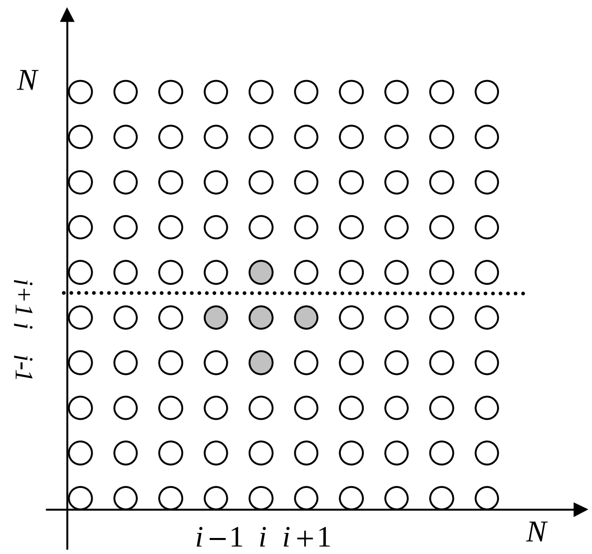


Fig.4.1 $N \times N$ grid points

It can be shown that the Jacobi method will eventually converge to the exact solution:

repeat

  forall $(1 \le i, j \le N)$

$$u_{i,j}^{m+1} = \frac{1}{4}(u_{i+1,j}^{m} + u_{i-1,j}^{m} + u_{i,j+1}^{m} + u_{i,j-1}^{m})$$

  endfor

until $u^{m+1} - u^{m}$ *small enough*

Where $u_{i,j}^{m+1}$ is the value of the grid point at (i, j) at iteration $m+1$. Each value in the solution is the average of the 4 nearest neighbors in the "old" grid. After each iteration we swap old and new grid points.

The computational domain is divided into a grid consisting of $N \times N$ points with the row-wise domain decomposition, where N is the number of grid points along a single dimension.

To update all grid points in one iteration, the time complexity on a single processor is the number of calculations (N^2) times the cost of a calculation (τ_{calc}). The time complexity can be described with the formula below:

$$T(1) = N^2 \tau_{calc} \qquad (4\text{-}6)$$

To update all grid points on p processors in one iteration, we have:

$$T(p) = \frac{N^2}{p}\tau_{calc} + T_{comm} \qquad (4\text{-}7)$$

Where T_{comm} is the total communication overhead in one iteration.

For a square domain, we have 3 decomposition strategies:

Case 1: Row-wise decomposition *Fig*.4.2(1*D*)

Case 2: Column-wise decomposition *Fig*.4.3(1*D*)

Case 3: Box-wise decomposition *Fig*.4.4(2*D*)

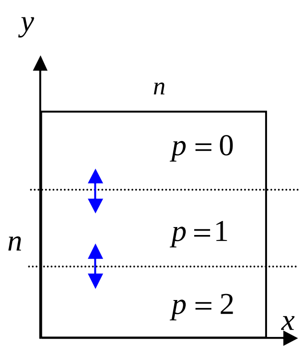


Fig .4.2 *Row-wise on one CE*

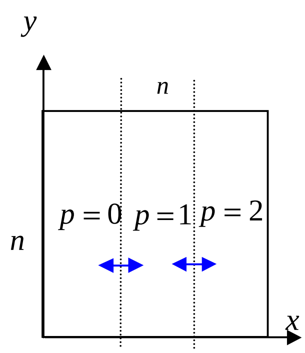


Fig.4.3 *Column-wise on one CE*

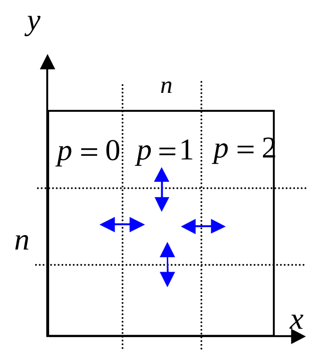


Fig.4.4 *Box-wise on one CE*

Case 1 and Case 2:

Case 1 and Case 2 differ a bit in programming algorithms. But in theory, they are really the same. Hence we discuss them together, but separate them in implementation.

In Eq.(4-8), T_{comm} , the communication cost, results from the point-to-point communication which consists of the latency of the network and the exchange time per data unit multiplied by the number of units communicated. In Fig.4.2, each processor has to exchange $2N$ borderline points with its upper neighbor and lower neighbor. (*Assume that there is a ring topology for all processors, meaning that it is periodic*).

Since we model the communication as point-to-point mode, by Eq.(3-3), we have:

$$T_{comm} = 2\left(\tau_{setup}^{cluster} + 2\times 8\times N\tau_{exchange}^{cluster}\right) \tag{4-8}$$

The coefficient 8 means 8 bytes per grid point in C language. Hence, we rewrite Eq.(4-8) and obtain a time complexity of:

$$T(p) = \left\lceil \frac{N}{p} \right\rceil N\tau_{calc} + 2\left(\tau_{setup}^{cluster} + 2\times 8\times N\tau_{exchange}^{cluster}\right) \tag{4-9}$$

(Notice that the imbalance of the workload has been taken into account in the equation (4-10), assuming $\frac{N}{p} \geq 2$, otherwise communication doesn't work.)

Speedup can be calculated by:

$$S(p) = \frac{T(1)}{T(p)} = \frac{N^2\tau_{calc}}{\left\lceil \frac{N}{p} \right\rceil N\tau_{calc} + 2\left(\tau_{setup}^{cluster} + 2\times 8\times N\tau_{exchange}^{cluster}\right)} \tag{4-10}$$

Comparing with Eq.(2-10) , we immediately realize that Eq.(4-10) is an explicit expression of Eq.(2-10) . Hence, we obtain:

$$\begin{aligned}
T_{1,p} &= \frac{n^2}{p\times\Delta} + 2T_{comm}^{cluster} \Leftrightarrow \\
T(p) &= \left\lceil \frac{N}{p} \right\rceil N\tau_{calc} + 2\left(\tau_{setup}^{cluster} + 2\times 8\times N\tau_{exchange}^{cluster}\right) \\
\tau_{calc} &= \frac{1}{\Delta} \\
T_{comm}^{cluster} &= \left(\tau_{setup}^{cluster} + 2\times 8\times N\tau_{exchange}^{cluster}\right)
\end{aligned} \tag{4-11}$$

Similarly, for C CEs , Eq.(2-11) can be interpreted as :

$$T_{C,p} = \frac{n^2}{pC\Delta} + 2T_{comm}^{grid} + 2T_{comm}^{cluster} \Leftrightarrow$$

$$T_{C,p} = \left\lceil \frac{N}{p} \right\rceil \frac{N}{C} \tau_{calc} + 2\left(\tau_{setup}^{grid} + 2\times 8\times N\tau_{exchange}^{grid}\right) + 2\left(\tau_{setup}^{cluster} + 2\times 8\times \frac{N}{C}\tau_{exchange}^{cluster}\right) \quad (4\text{-}12)$$

$$T_{comm}^{grid} = \left(\tau_{setup}^{grid} + 2\times 8\times N\tau_{exchange}^{grid}\right)$$

Here, we assume that there exist more than 2 clusters and not less than 3 processors in each cluster. Substituting Eq.(2-14) with Eq.(4-12) and Eq.(4-13), we obtain the grid speedup:

$$G_p^C = \frac{T_{1,p}}{T_{C,p}} = \frac{C}{1 + \frac{2C\alpha}{\beta + 2}} \quad (4\text{-}13)$$

Where

$$\alpha = \frac{T_{comm}^{grid}}{T_{comm}^{cluster}} = \frac{\tau_{setup}^{grid} + 2\times 8\times N\tau_{exchange}^{grid}}{\tau_{setup}^{cluster} + 2\times 8\times N\tau_{exchange}^{cluster}} \quad (4\text{-}14)$$

$$\beta = \frac{N^2}{p\left(\tau_{setup}^{cluster} + 2\times 8\times N\tau_{exchange}^{cluster}\right)}\tau_{calc} \quad (4\text{-}15)$$

So far, α and β have explicit expressions. We will take a close look at 2 parameters.

For parameter α:

If problem size $N \to +\infty$, then $\alpha \to \frac{\tau_{exchange}^{grid}}{\tau_{exchange}^{cluster}} = \frac{85}{5} = 17$

And as mentioned before, $N/p \geq 2$ and $p \geq 2$, then we have:

$$\alpha \to \frac{\tau_{setup}^{grid} + 2\times 8\times N\tau_{exchange}^{grid}}{\tau_{setup}^{cluster} + 2\times 8\times N\tau_{exchange}^{cluster}} = \frac{1600000 + 2\times 8\times 4\times 85}{8000 + 2\times 8\times 4\times 5} = 199.78$$

Hence we conclude that parameter α is over $(17, 199.78]$.

For parameter β:

If problem size $N \to +\infty$, then $\beta \to +\infty$.

If $N/p \geq 2$ and $p \geq 2$, then $\beta \to 0.59$

So parameter β is over $[0.59,+\infty)$.

Case 3:

Fig.4.4 shows the box-wise decomposition, which is 2D decomposition along 2 dimensions (x and y in Cartesian coordinate) . In this case, each processor sends and receives four pairs of ghost cell data ($\frac{N}{\sqrt{p}} \geq 2$), coming from the four nearest neighboring processors. The time complexity for the box-wise decomposition is:

$$T(1) = N^2 \tau_{calc} \tag{4-16}$$

$$T_{comm} = 4\left(\tau_{setup}^{cluster} + 2\times 8\times \frac{N}{\sqrt{p}} \tau_{exchange}^{cluster}\right) \tag{4-17}$$

$$T(p) = \left\lceil \frac{N}{p} \right\rceil N\tau_{calc} + 4\left(\tau_{setup}^{cluster} + 2\times 8\times \frac{N}{\sqrt{p}} \tau_{exchange}^{cluster}\right) \tag{4-18}$$

Assume that p is a perfect square integer and $\sqrt{p}$ divides N evenly. We are interested in comparing the performance of Eq.(4-9) and (4-18). The difference of Eq.(4-9) and (4-18) is only from the term T_{comm}. Therefore, Eq. (4-18) minus Eq. (4-9) is given:

$$2\tau_{setup}^{cluster} + \frac{64\, N\, \tau_{exchange}^{cluster}}{\sqrt{p}} - 32\, N\, \tau_{exchange}^{cluster} < 0 \tag{4-19}$$

Let the expression (4-20) be less than 0 and after several arithmetic conversions we have:

$$\frac{\tau_{setup}^{cluster}}{\tau_{exchange}^{cluster}} \frac{\sqrt{p}}{(16\sqrt{p} - 32)} < N \tag{4-20}$$

In Eq.(4-20), the values of p should be more than 4. Let us estimate the value of N if p= 16. We use the values of the parameters in Table-3.1 to fill Eq.(4-20) without consideration of standard deviation. We find the box-wise decomposition will outperform the row-wise only when N>200. Table-4.1 shows the empirical results with standard deviations. Each data in Table-4.1 averaged a sample of 10.

Table – 4.1 Comparisons of Box-wise and Row-wise (One iteration)

N (problem size)	200	256	400	512	768
Box-wise($10^{-4}s$)	2.74±0.01	3.78±0.02	9.48±1.13	19.3±1.32	61.8±2.30
Row-wise($10^{-4}s$)	2.69±0.05	3.81±0.09	9.87±0.95	20.3±1.14	79.7±6.30

4.2. Experiment Setups

In this subsection, we discuss two important issues before we are starting to implement our experiments. One is topology and another is workload balance. Because MPI is very sensitive to topologies, we have to explicitly preset suitable topology in our code. We also consider how to distribute workload among clusters as evenly as possible since workload imbalance also extremely affects on empirical results.

4.2.1 Cartesian Topology

A virtual topology is a mechanism for assigning IDs to all processors in a communicator in a way that fits communication patterns better. MPI, for portability, consciously hides the hardware communication architecture from users. Hence, we in fact work abstractly with MPI. However for an executable application, we should know how to map process ranks onto coordinates in a grid.

To achieve the mapping between process ranks and coordinates after the fashion of a 2D grid, we have to create a virtual topology and then use MPI mapping functions to compute ranks, based on the given naming scheme.

MPI provides 2 virtual topologies: one is Cartesian topology and another is Graph topology. In a Cartesian topology, each process is connected to its neighbors in a virtual grid. Whereas, in a Graph topology, each process is maybe connected to any number of other processes and the numbering is arbitrary.

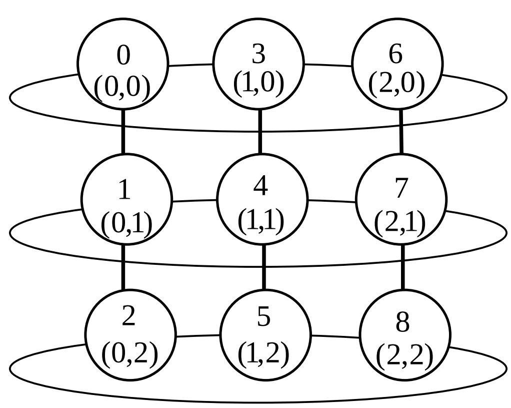


Fig.4.5 Periodic boundary in x direction

Using a virtual topology might gain us performance benefit such as optimized communication patterns and programming simplifications.

As with everything else in MPI, a virtual topology is associated with a communicator, which groups several processes. Once a new communicator is created on an existing communicator, the old one is immediately replaced by the new one and of course the new one will use the same virtual topology as the old did.

In our cases, we come to realize that a 2D Cartesian topology is suitable for our implementation. The Cartesian grid can be of any dimension and may be periodic or not in any dimension. Furthermore, tori, rings and 3D grids are all supported. Fig.4.5 shows a 2D Cartesian grid with 9 processes in which each processor is assigned to a unique ID and in x direction the boundary condition is cyclic, corresponding to the initial conditions of the Laplace equation we explained in the previous section.

In case 1 Fig.4.2, if a workload in level 1 is decomposed as in level 2 in row-wise, the information returned by MPI will be depicted in Fig.4.6. For case 2, Fig.4.7 shows the resulting information. For case 3, the results are shown as in Fig.4.8.

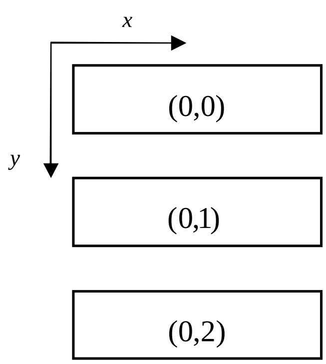


Fig.4.6 *Level* 1 *and Level* 2 *in row-wise*

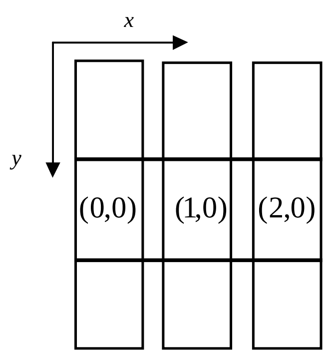


Fig.4.7 *Level* 1 *in row-wise* *Level* 2 *in column-wise*

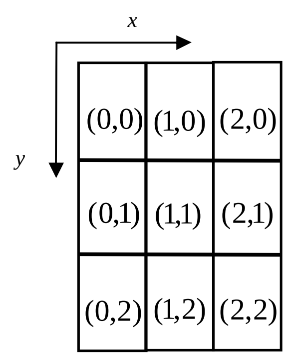


Fig.4.8 *Level* 1 *in row-wise* *Level*2 *in box-wise*

In our implementations, we chose 3 algorithms for three cases as similar as possible.

4.2.2 Workload Balance

The workload balance can decrease communication overheads. Ideally, each processor locally works with the same portion of the total workload. But, in practice, it is quite difficult to decompose a workload perfectly. So we only can decompose a workload as evenly as possible. Usually, only one processor receives an extra portion. More details can be found from debugging messages after running our parallel applications.

4.3. Results

4.3.1 Theoretical and Empirical Results for Case I and Case II

This subsection shows empirical results comparing with theoretical ones. Fig.4.9 shows theoretical grid speedups and efficiencies of Case I and Case II concerned with 2 and 3 CEs. Fig4.10 shows the analytical results of Case III on 4 CEs. Each experiment took a sample of 10 since sometimes network does not work very well. Fig.4.11 illustrates empirical grid speedups as functions of the ratio of problem size to the number of processors. From Fig.4.9, Fig.4.10 and Fig.4.11, we can conclude that the empirical grid speedups approximate very well to analytical ones and also proved that Grid Speedup theory is correct and feasible in a real grid environment.

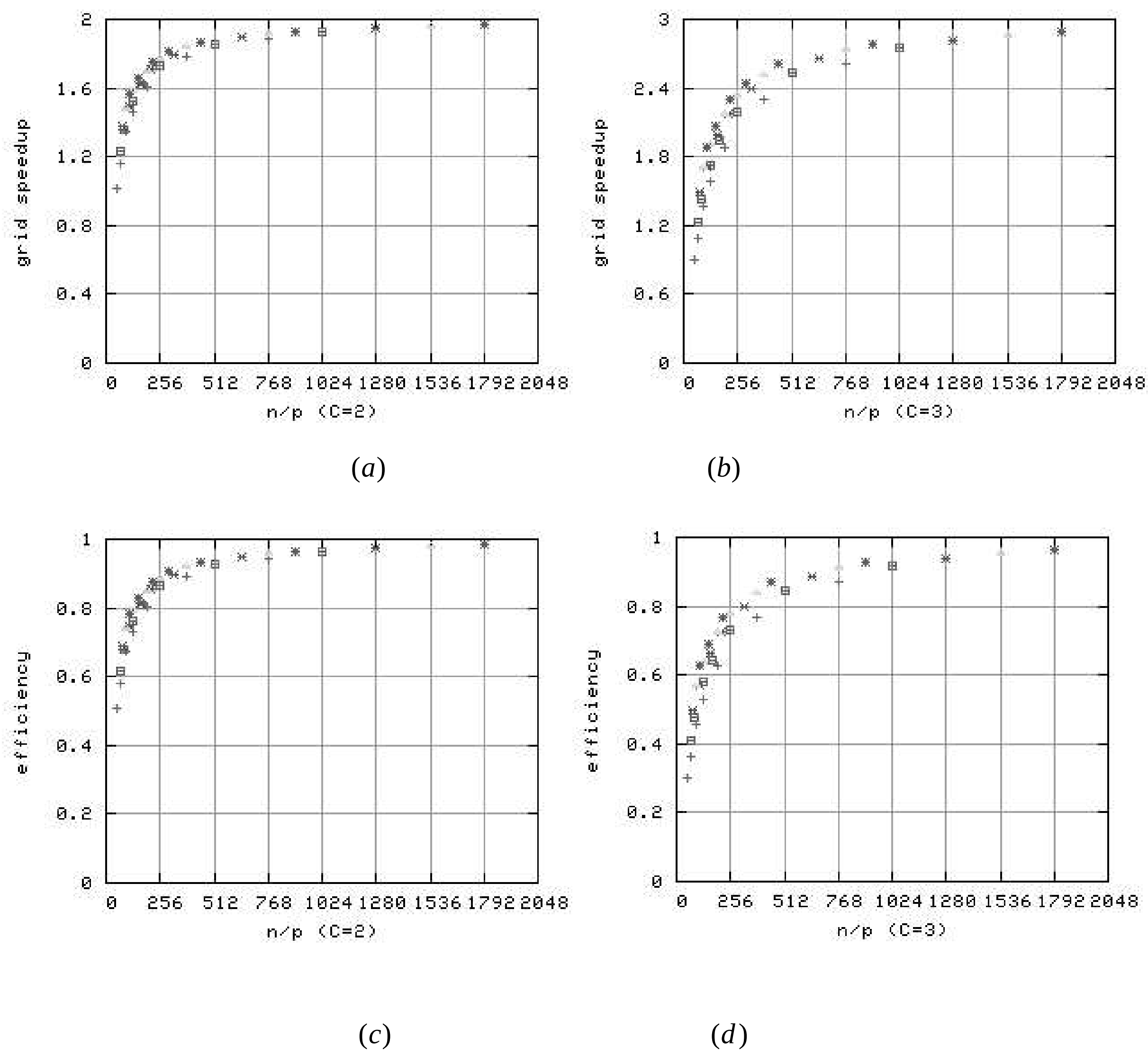
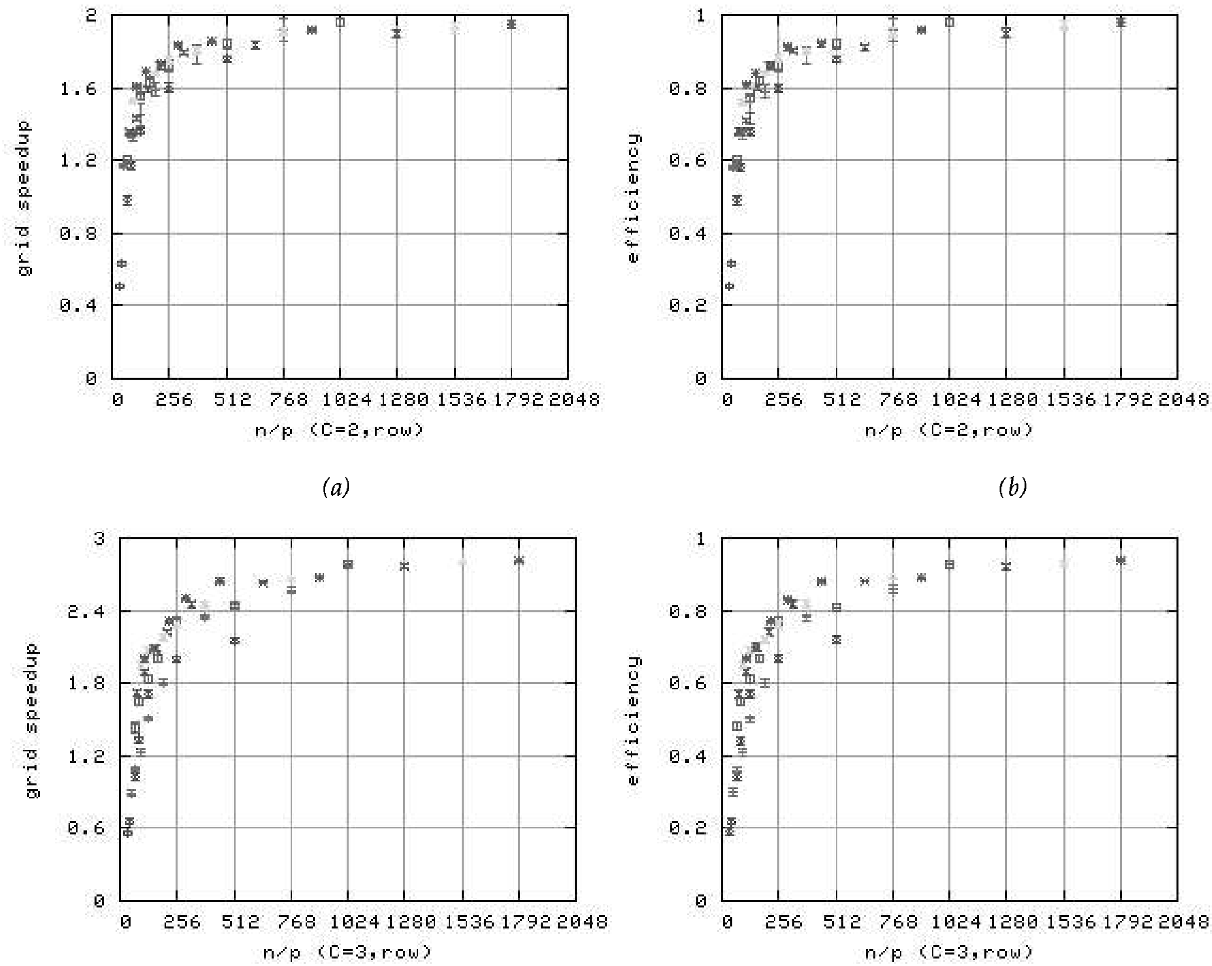


Fig.4.9 Theoretical grid speedups and efficiencies for Case I and Case II

(a) Grid speedups on 2 CEs *(b) Grid speedups on 3 CEs*

(c) Grid efficiencies on 2 CEs *(d) Grid efficiencies on 3 CEs*

In Fig.4.9, graph (a), (b), (c) and (d) plot the analytical results for Case I and Case II. As discussed in previous sections, the analytical results of Case I and Case II are the same.



(c) *(d)*

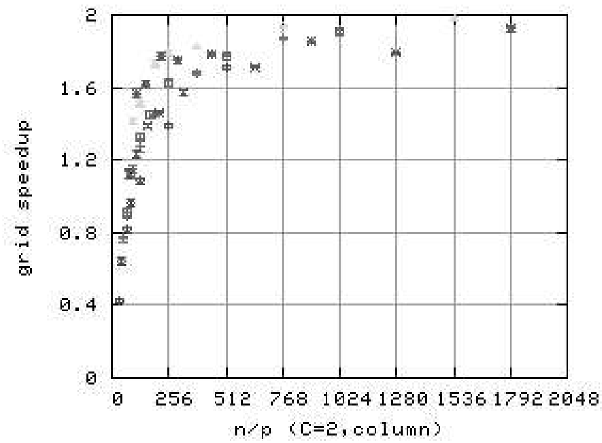
Fig.4.10 Emperical Results for Case I

(a) Grid Speedups on 2 CEs

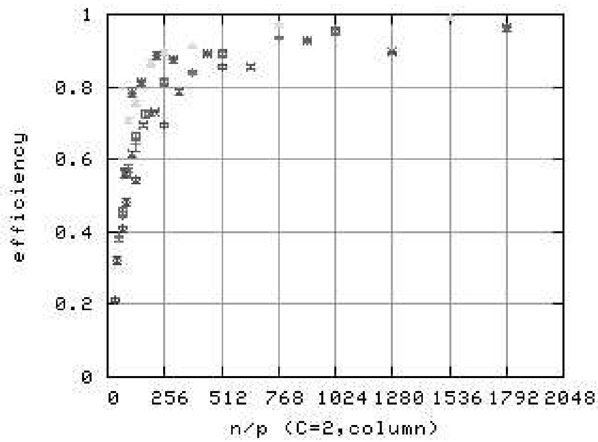
(b) Grid Efficiencies on 3 CEs

(c) Grid Speedups on 2 CEs

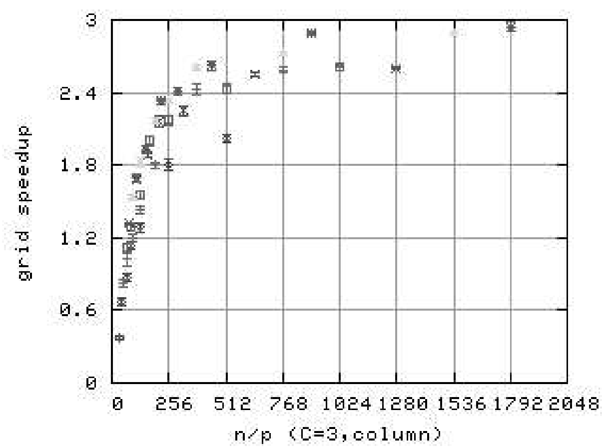
(d) Grid Efficiencies on 3 CEs



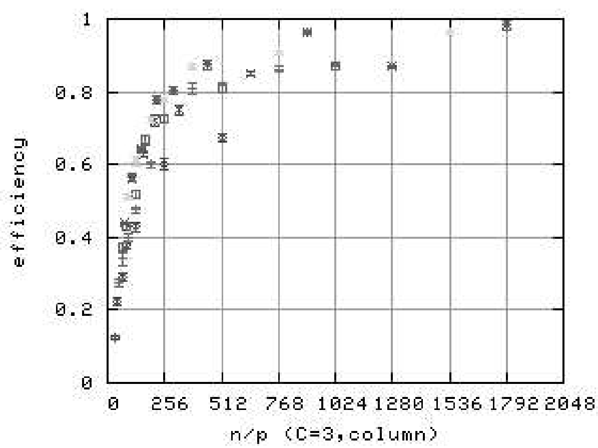
(a)



(b)



(c)

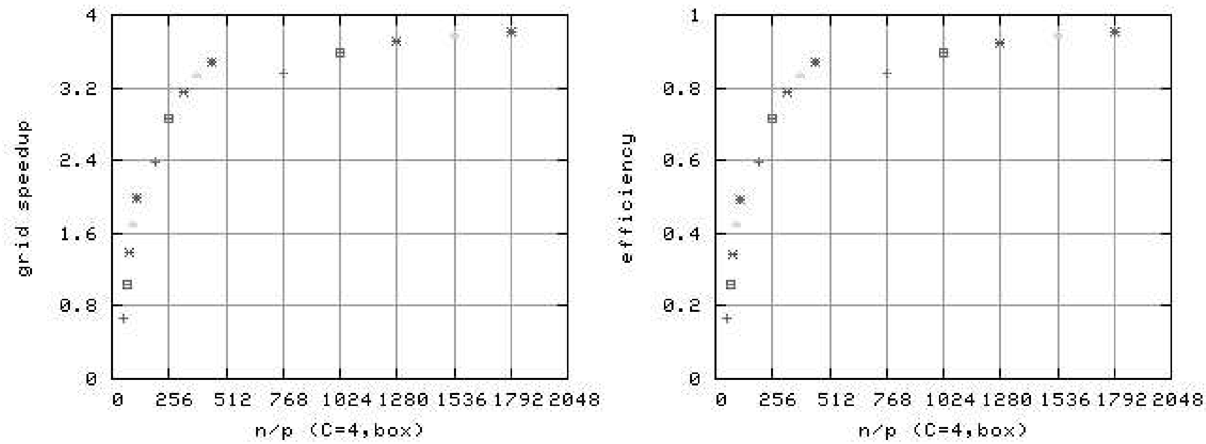


(d)

Fig.4.11 Emperical Results for Case II

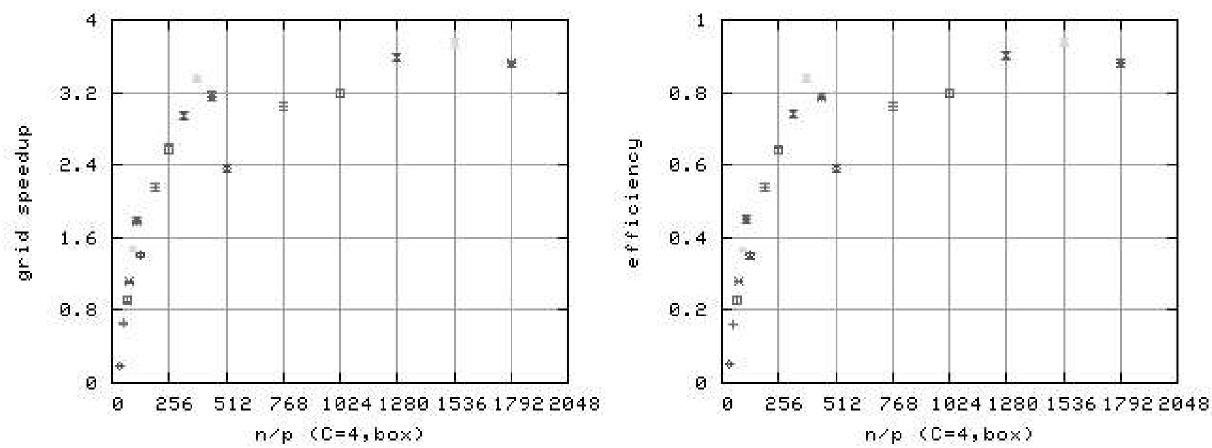
(a) Grid Speedups on 2 CEs
(b) Grid Efficiencies on 3 CEs
(c) Grid Speedups on 2 CEs
(d) Grid Efficiencies on 3 CEs

4.3.2 Theoretical and Empirical Results Case III



(a) *(b)*

Fig.4.12 Theoretical Grid Speedups and Efficiencies for Case III on 4 CEs
(a) Grid Speedups (b) Grid Efficiencies



(a) *(b)*

Fig.4.13 Emperical Grid Speedups and Efficiencies for Case III on 4 CEs
(a) Grid Speedups (b) Grid Efficiencies

4.4 Summary

In this section, we validated Grid Speedup theory with experiments. Through the comparisons of analytical and empirical results, we can conclude that Grid Speedup theory is correct for the extension on Amdahl's law so far.

Chapter V Conclusions and Directions for Future Work

In this chapter, we will discuss the work we have finished so far and conclude our work. Consecutively, we will recommend some directions for future work in Grid Speedup theory.

5.1 Conclusions

Earlier chapters have already shown that the Grid Speedup theory has been successfully used in a real grid environment. Though we validated Grid Speedup theory with a simple application as a benchmarking program, we have to point out that some limitations inherent from our original model make our experiments imperfect and incomplete. Firstly, FDM (Finite Difference Method) is only suitable for regular geometry such as squares, rectangles and cubes, not for complex geometry like spheres and cylinders. Secondly, our mathematical model based on 2D Laplace equation is so symmetric that it is not completed representative. All in all, we can conclude that Grid Speedup theory is a success for performance metrics in HPC community.

5.2 Directions for Future Work

The time for this project has used up and we have to leave some work for future. Some directions are far more likely to exist in the flowing aspects.

5.2.1 Benchmarking Programs

More versatile applications as benchmarking programs will be chosen in hope of refining the Grid Speedup theory. There have been many applications available and suitable to validate the Grid Speedup theory like DLA (Diffusion-Limited Aggregations), LBM(Lattice Boltzmann Method) BGK and NBody. We may try to use different applications with various mathematical techniques like FEM (Finite Element Method) and FVM (Finite Volume Method) as well as FDM. Only after "hammer-hardened" by using various applications in a real grid environment, Grid Speedup theory will not go in its wrong perspective.

5.2.2 Extensions On Grid Speedup Theory

Hoekstra [Hoekstra05] also represents a special case, where C is a fractional number, not an integer. We can interpret it as an asymmetrical case, in which all processors in some CEs are fully involved in computing and some CEs are not. We will implement this special case in future.

In previous implementation, there are in total 4 clusters available. Our experiments with Case 3 could not be conducted sufficiently. We will validate Grid Speedup theory using more than 9 clusters in our future work.

Appendix I Theoretical Data

Table-A01 Theoretical Grid Speedups on 2 CEs
(Case 1 and Case 2)

n/p	G (Grid speedups ,C=2)					
32	0.701					
43	0.832					
48		1.017				
64	1.028	1.156	1.233			
80				1.381		
86	1.168		1.362			
96		1.343			1.487	
107				1.496		
112						1.564
128	1.354	1.462	1.522		1.588	
150						1.654
160				1.632		
171			1.618			
192		1.605			1.704	
214				1.710		
224						1.754
256	1.612		1.727		1.769	
299						1.810
320				1.797		
384		1.780			1.840	
448						1.869
512	1.785		1.853			
640				1.893		
768		1.883			1.916	
896						1.932
1024			1.924			
1280				1.945		
1536					1.957	
1792						1.965

Table-A02 Theoretical Grid Speedups on 3 CEs

(Case 1 and Case 2)

n/p	G (Grid Speedups ,C=3)					
32	0.525					
43	0.656					
48		0.897				
64	0.881	1.084	1.224			
80				1.492		
86	1.067		1.434			
96		1.372			1.706	
107				1.704		
112						1.877
128	1.355	1.585	1.733		1.910	
150						2.069
160				1.988		
171			1.936			
192		1.879			2.172	
214				2.170		
224						2.306
256	1.861		2.194		2.332	
299						2.447
320				2.390		
384		2.308			2.518	
448						2.607
512	2.296		2.533			
640				2.660		
768		2.608			2.738	
896						2.789
1024			2.747			
1280				2.819		
1536					2.863	
1792						2.891

Table-A03 Theoretical Grid efficiencies on 2 CEs

(Case 1 and Case 2)

n/p	γ (Grid efficiencies, C=2)					
32	0.351					
43	0.416					
48		0.508				
64	0.514	0.578	0.616			
80				0.691		
86	0.584		0.681			
96		0.671			0.743	
107				0.748		
112						0.782
128	0.677	0.731	0.761		0.794	
150						0.827
160				0.816		
171			0.809			
192		0.802			0.852	
214				0.855		
224						0.877
256	0.806		0.864		0.885	
299						0.905
320				0.898		
384		0.890			0.920	
448						0.934
512	0.892		0.927			
640				0.946		
768		0.942			0.958	
896						0.966
1024			0.962			
1280				0.972		
1536					0.979	
1792						0.983

Table-A04 Theoretical Grid efficiencies on 3 CEs
(Case 1 and Case 2)

n/p	γ (Grid efficiencies, C=3)					
32	0.175					
43	0.219					
48		0.299				
64	0.294	0.361	0.408			
80				0.497		
86	0.356		0.478			
96		0.457			0.569	
107				0.568		
112						0.626
128	0.452	0.528	0.578		0.637	
150						0.690
160				0.663		
171			0.645			
192		0.626			0.724	
214				0.723		
224						0.769
256	0.620		0.731		0.777	
299						
320				0.797		
384		0.769			0.839	
448						0.869
512	0.765		0.844			
640				0.887		
768		0.869			0.913	
896						0.930
1024			0.916			
1280				0.940		
1536					0.954	
1792						0.964

Table-A05 Theoretical Grid Speedups on 4 CEs

(Case 3)

n/p	G (Grid speedups ,C=4)					
32	0.340					
48		0.670				
64			1.029			
80				1.378		
96					1.697	
112						1.977
128	1.631					
192		2.379				
256			2.851			
320				3.149		
384					3.344	
448						3.477
512	2.856					
768		3.348				
1024			3.575			
1280				3.697		
1536					3.769	
1792						3.817

Table-A06 Theoretical Grid efficiencies on 4 CEs

(Case 3)

n/p	γ (Grid efficiencies, C=4)					
32	0.085					
48		0.167				
64			0.257			
80				0.344		
96					0.424	
112						0.494
128	0.408					
192		0.595				
256			0.713			
320				0.787		
384					0.836	
448						0.869
512	0.714					
768		0.837				
1024			0.894			
1280				0.924		
1536					0.942	
1792						0.954

Appendix II Experimental Data
(Case 1)

We used the following formulas to process the experimental datasets.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \qquad \sigma^2 = \frac{1}{N-1}\left(\sum_{i=1}^{N}(x_i)^2 - \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N}\right)$$

$$if\ \ x = \bar{x} \pm \sigma_x, and\ y = \bar{y} \pm \sigma_y,\ \ then\ \ z = \frac{x}{y} = \frac{\bar{x}}{\bar{y}} \pm \sigma_z$$

$$StdDev = \sigma_z = \pm\frac{\bar{x}}{\bar{y}}\sqrt{\left(\frac{\sigma_x}{\bar{x}}\right)^2 + \left(\frac{\sigma_y}{\bar{y}}\right)^2}$$

Where N is the size of a sample, $\bar{x}$ and $\bar{y}$ are mean values, σ is standard deviation.

Table – A07 Empirical Grid Speedups on 2 CEs

n/p	G	±StdDev	G	±StdDev	G	±StdDev	G	±StdDev	G	±StdDev	G	±StdDev
32	0.509	0.011										
43	0.630	0.012										
48			1.167	0.013								
64	0.981	0.027	1.179	0.016	1.195	0.006						
80							1.363	0.014				
86	1.165	0.021			1.352	0.010						
96			1.338	0.028					1.516	0.014		
107							1.429	0.009				
112											1.610	0.006
128	1.357	0.018	1.452	0.060	1.548	0.016			1.614	0.007		
150											1.689	0.003
160							1.590	0.016				
171					1.632	0.023						
192			1.588	0.035					1.683	0.013		
214							1.722	0.021				
224											1.726	0.008
256	1.597	0.023			1.722	0.034			1.752	0.021		
299											1.829	0.012
320							1.793	0.012				
384			1.783	0.049					1.800	0.016		
448											1.845	0.014
512	1.755	0.017			1.836	0.026						
640							1.829	0.016				
768			1.919	0.064					1.905	0.024		
896											1.917	0.011
1024					1.964	0.019						
1280							1.895	0.024				
1536									1.934	0.027		
1792											1.950	0.017

Table – A08 Empirical Grid Speedups on 3 CEs

n/p	G	±StdDev	G	±StdDev	G	±StdDev	G	±StdDev	G	±StdDev	G	±StdDev
32	0.560	0.023										
43	0.663	0.019										
48			0.885	0.022								
64	1.027	0.030	1.088	0.016	1.434	0.039						
80							1.722	0.018				
86	1.330	0.026			1.649	0.028						
96			1.229	0.027					1.950	0.030		
107							1.887	0.018				
112											1.996	0.015
128	1.707	0.027	1.512	0.024	1.835	0.028			2.068	0.030		
150											2.085	0.025
160							2.087	0.018				
171					2.010	0.028						
192			1.812	0.019					2.169	0.030		
214							2.221	0.018				
224											2.311	0.009
256	1.998	0.015			2.309	0.017			2.274	0.017		
299											2.503	0.009
320							2.449	0.018				
384			2.345	0.024					2.447	0.016		
448											2.653	0.016
512	2.146	0.022			2.436	0.012						
640							2.631	0.010				
768			2.574	0.024					2.663	0.012		
896											2.683	0.009
1024					2.781	0.012						
1280							2.771	0.018				
1536									2.800	0.019		
1792											2.814	0.009

Table – A09 Empirical Grid Efficiencies on 2 CEs

n/p	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev
32	0.255	0.006										
43	0.315	0.006										
48			0.584	0.007								
64	0.491	0.014	0.590	0.008	0.598	0.003						
80							0.682	0.007				
86	0.583	0.011			0.676	0.005						
96			0.669	0.014					0.758	0.007		
107							0.715	0.005				
112											0.805	0.003
128	0.679	0.009	0.726	0.030	0.774	0.008			0.807	0.004		
150											0.845	0.002
160							0.795	0.008				
171					0.816	0.012						
192			0.794	0.018					0.842	0.007		
214							0.861	0.011				
224											0.863	0.004
256	0.799	0.012			0.861	0.017			0.876	0.011		
299											0.915	0.006
320							0.897	0.006				
384			0.892	0.025					0.900	0.008		
448											0.923	0.007
512	0.878	0.009			0.918	0.013						
640							0.915	0.008				
768			0.960	0.032					0.953	0.012		
896											0.959	0.006
1024					0.982	0.010						
1280							0.948	0.012				
1536									0.967	0.014		
1792											0.975	0.009

Table – A10 Empirical Grid Efficiencies on 3 CEs

n/p	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev
32	0.187	0.008										
43	0.221	0.006										
48			0.295	0.007								
64	0.342	0.010	0.363	0.005	0.478	0.013						
80							0.574	0.006				
86	0.443	0.009			0.550	0.009						
96			0.410	0.009					0.650	0.010		
107							0.629	0.006				
112											0.665	0.005
128	0.569	0.009	0.504	0.008	0.612	0.009			0.689	0.010		
150											0.695	0.008
160							0.696	0.006				
171					0.670	0.009						
192			0.604	0.006					0.723	0.010		
214							0.740	0.006				
224											0.770	0.003
256	0.666	0.005			0.770	0.006			0.758	0.006		
299											0.834	0.003
320							0.816	0.006				
384			0.782	0.008					0.816	0.005		
448											0.884	0.005
512	0.715	0.007			0.812	0.004						
640							0.877	0.003				
768			0.858	0.008					0.888	0.004		
896											0.894	0.003
1024					0.927	0.004						
1280							0.924	0.006				
1536									0.933	0.006		
1792											0.938	0.003

$Table - A11\ \ Experimental\ \ Data\ on\ 2\ CES\ (Case\ 1)$

n=512, C=1					n=512,C=2				Results					
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	StdDev	ana_512	exp_512	$\pm\sigma_z$
1	931.802	868254.967	4.960	24.602	530.995	281955.690	1.946	3.787	1.755	0.011	1.785	1.755	0.011
2	674.916	455511.607	2.712	7.355	422.732	178702.344	2.678	7.172	1.597	0.012	1.612	1.597	0.012
4	345.582	119426.919	4.445	19.758	254.632	64837.455	3.890	15.132	1.357	0.027	1.354	1.357	0.027
6	221.816	49202.338	2.915	8.497	190.320	36221.702	2.412	5.818	1.165	0.021	1.168	1.165	0.021
8	163.020	26575.520	1.955	3.822	166.173	27613.466	2.317	5.368	0.981	0.018	1.028	0.981	0.018
12	108.794	11836.134	2.152	4.631	172.804	29861.222	5.212	27.165	0.630	0.023	0.832	0.630	0.023
16	78.661	6187.553	2.007	4.028	154.654	23917.860	3.457	11.951	0.509	0.017	0.701	0.509	0.017
n=768, C=1					n=768,C=2				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	StdDev	ana_768	exp_768	$\pm\sigma_z$
1	2143.692	4595415.391	4.344	18.870	1117.032	1247760.489	7.178	51.524	1.919	0.013	1.883	1.919	0.013
2	1507.314	2271995.495	8.046	64.738	845.597	715034.286	5.853	34.258	1.783	0.016	1.780	1.783	0.016
4	644.979	415997.910	1.859	3.456	406.112	164926.957	7.172	51.438	1.588	0.028	1.605	1.588	0.028
6	441.004	194484.528	4.148	17.206	303.782	92283.504	3.285	10.791	1.452	0.021	1.462	1.452	0.021
8	343.998	118334.624	3.071	9.431	257.045	66072.132	6.239	38.925	1.338	0.035	1.343	1.338	0.035
12	232.698	54148.359	1.473	2.170	197.239	38903.223	3.951	15.610	1.180	0.025	1.156	1.180	0.025
16	201.621	40651.028	1.853	3.434	172.800	29859.840	2.045	4.182	1.167	0.017	1.017	1.167	0.017
n=1024, C=1					n=1024,C=2				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	StdDev	ana_1024	exp_1024	$\pm\sigma_z$
1	3661.113	13403748.399	6.203	38.477	1863.972	3474391.617	5.084	25.847	1.964	0.006	1.924	1.964	0.006
2	2355.317	5547518.170	5.087	25.878	1282.510	1644831.900	6.382	40.730	1.836	0.010	1.853	1.836	0.010
4	1254.730	1574347.373	4.744	22.506	728.688	530986.201	6.228	38.788	1.722	0.016	1.727	1.722	0.016
6	870.576	757902.572	10.833	117.354	533.289	284397.158	3.740	13.988	1.632	0.023	1.618	1.632	0.023
8	641.983	412142.172	4.945	24.453	414.727	171998.485	8.639	74.632	1.548	0.034	1.522	1.548	0.034
12	466.936	218029.228	4.967	24.671	345.281	119218.969	5.460	29.812	1.352	0.026	1.362	1.352	0.026
16	340.622	116023.347	4.003	16.024	284.983	81215.310	3.009	9.054	1.195	0.019	1.233	1.195	0.019

(Continue…)

	n=1280, C=1				n=1280,C=2				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	StdDev	ana_1280	exp_1280	$\pm\sigma_z$
1	5987.461	35849689.227	33.633	1131.179	3160.046	9985890.722	14.636	214.212	1.895	0.014	1.945	1.895	0.014
2	4159.733	17303378.631	12.031	144.745	2274.113	5171589.937	8.312	69.089	1.829	0.009	1.893	1.829	0.009
4	2143.756	4595689.788	12.286	150.946	1195.797	1429930.465	8.239	67.881	1.793	0.016	1.797	1.793	0.016
6	1390.128	1932455.856	6.587	43.389	807.388	651875.383	9.170	84.089	1.722	0.021	1.710	1.722	0.021
8	1055.103	1113242.341	3.458	11.958	663.460	440179.172	4.456	19.856	1.590	0.012	1.632	1.590	0.012
12	729.637	532370.152	5.525	30.526	510.526	260636.797	4.361	19.018	1.429	0.016	1.496	1.429	0.016
16	555.868	308989.233	4.467	19.954	407.820	166317.152	6.483	42.029	1.363	0.024	1.381	1.363	0.024
	n=1536, C=1				n=1536,C=2				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	StdDev	ana_1536	exp_1536	$\pm\sigma_z$
1	8694.286	75590609.050	59.461	3535.611	4495.306	20207776.034	9.482	89.908	1.934	0.014	1.957	1.934	0.014
2	6026.705	36321173.157	19.154	366.876	3164.396	10013402.045	6.660	44.356	1.905	0.007	1.916	1.905	0.007
4	2693.170	7253164.649	7.088	50.240	1496.081	2238258.359	10.268	105.432	1.800	0.013	1.840	1.800	0.013
6	1865.266	3479217.251	8.983	80.694	1064.430	1133011.225	11.513	132.549	1.752	0.021	1.769	1.752	0.021
8	1420.884	2018911.341	10.458	109.370	844.204	712680.394	5.327	28.377	1.683	0.016	1.704	1.683	0.016
12	999.424	998848.332	6.823	46.553	619.202	383411.117	8.125	66.016	1.614	0.024	1.588	1.614	0.024
16	771.131	594643.019	5.542	30.714	508.597	258670.908	8.358	69.856	1.516	0.027	1.487	1.516	0.027
	n=1792, C=1				n=1792,C=2				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	StdDev	ana_1792	exp_1792	$\pm\sigma_z$
1	11955.905	142943664.369	30.074	904.445	6132.373	37605998.611	10.530	110.881	1.950	0.006	1.965	1.950	0.006
2	8164.715	66662571.031	10.121	102.435	4259.334	18141926.124	4.881	23.824	1.917	0.003	1.932	1.917	0.003
4	4202.467	17660728.886	12.489	155.975	2277.907	5188860.301	7.415	54.982	1.845	0.008	1.869	1.845	0.008
6	2903.388	8429661.879	6.421	41.229	1587.607	2520495.986	9.558	91.355	1.829	0.012	1.810	1.829	0.012
8	2067.048	4272687.434	7.440	55.354	1197.464	1433920.031	8.354	69.789	1.726	0.014	1.754	1.726	0.014
12	1563.484	2444482.218	6.316	39.892	925.867	857229.702	4.465	19.936	1.689	0.011	1.654	1.689	0.011
16	1135.359	1289040.059	7.864	61.842	705.096	497160.369	5.591	31.259	1.610	0.017	1.564	1.610	0.017

Table – A12 Experimental Data on 3 CES (Case 1)

n=512, C=1					n=512,C=3				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_512	exp_512	$\pm\sigma_z$
1	931.802	868254.967	4.960	24.602	434.240	188564.378	3.948	15.587	2.146	0.023	2.296	2.146	0.023
2	674.916	455511.607	2.712	7.355	337.835	114132.487	2.913	8.486	1.998	0.019	1.861	1.998	0.019
4	345.582	119426.919	4.445	19.758	202.486	41000.580	2.409	5.803	1.707	0.030	1.355	1.707	0.030
6	221.816	49202.338	2.915	8.497	166.773	27813.234	2.485	6.175	1.330	0.026	1.067	1.330	0.026
8	163.020	26575.520	1.955	3.822	158.741	25198.705	3.685	13.579	1.027	0.027	0.881	1.027	0.027
12	108.794	11836.134	2.152	4.631	164.125	26937.016	1.658	2.749	0.663	0.015	0.656	0.663	0.015
16	78.661	6187.553	2.007	4.028	140.568	19759.363	4.098	16.794	0.560	0.022	0.525	0.560	0.022
n=768, C=1					n=768,C=3				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_768	exp_768	$\pm\sigma_z$
1	2143.692	4595415.391	4.344	18.870	832.873	693677.434	6.806	46.322	2.574	0.022	2.608	2.574	0.022
2	1507.314	2271995.495	8.046	64.738	642.757	413136.561	2.737	7.491	2.345	0.016	2.308	2.345	0.016
4	644.979	415997.910	1.859	3.456	356.019	126749.528	3.614	13.061	1.812	0.019	1.879	1.812	0.019
6	441.004	194484.528	4.148	17.206	291.728	85105.226	3.733	13.935	1.512	0.024	1.585	1.512	0.024
8	343.998	118334.624	3.071	9.431	279.869	78326.657	5.738	32.925	1.229	0.027	1.372	1.229	0.027
12	232.698	54148.359	1.473	2.170	213.853	45733.106	3.175	10.081	1.088	0.018	1.084	1.088	0.018
16	201.621	40651.028	1.853	3.434	227.821	51902.408	3.103	9.629	0.885	0.015	0.897	0.885	0.015
n=1024, C=1					n=1024,C=3				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1024	exp_1024	$\pm\sigma_z$
1	3661.113	13403748.399	6.203	38.477	1316.392	1732887.898	3.675	13.506	2.781	0.009	2.747	2.781	0.009
2	2355.317	5547518.170	5.087	25.878	967.019	935125.746	4.226	17.859	2.436	0.012	2.533	2.436	0.012
4	1254.730	1574347.373	4.744	22.506	543.314	295190.103	3.506	12.292	2.309	0.017	2.194	2.309	0.017
6	870.576	757902.572	10.833	117.354	433.052	187534.035	2.943	8.661	2.010	0.029	1.936	2.010	0.029
8	641.983	412142.172	4.945	24.453	349.911	122437.708	3.886	15.101	1.835	0.025	1.733	1.835	0.025
12	466.936	218029.228	4.967	24.671	283.124	80159.199	3.722	13.853	1.649	0.028	1.434	1.649	0.028
16	340.622	116023.347	4.003	16.024	237.542	56426.202	5.907	34.893	1.434	0.039	1.224	1.434	0.039

(Continue…)

n=1280, C=1					n=1280,C=3				Results				
p	$\overline{x}$	$\left(\overline{x}\right)^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$\left(\overline{y}\right)^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1280	exp_1280	$\pm\sigma_z$
1	5987.461	35849689.227	33.633	1131.179	2160.904	4669506.097	5.458	29.790	2.771	0.017	2.819	2.771	0.017
2	4159.733	17303378.631	12.031	144.745	1580.855	2499102.531	4.826	23.290	2.631	0.011	2.660	2.631	0.011
4	2143.756	4595689.788	12.286	150.946	875.478	766461.728	4.022	16.176	2.449	0.018	2.390	2.449	0.018
6	1390.128	1932455.856	6.587	43.389	626.037	391922.325	4.318	18.645	2.221	0.019	2.170	2.221	0.019
8	1055.103	1113242.341	3.458	11.958	505.561	255591.925	5.041	25.412	2.087	0.022	1.988	2.087	0.022
12	729.637	532370.152	5.525	30.526	386.612	149468.839	3.000	9.000	1.887	0.020	1.704	1.887	0.020
16	555.868	308989.233	4.467	19.954	322.869	104244.391	2.978	8.868	1.722	0.021	1.492	1.722	0.021
n=1536, C=1					n=1536,C=3				Results				
p	$\overline{x}$	$\left(\overline{x}\right)^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$\left(\overline{y}\right)^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1536	exp_1536	$\pm\sigma_z$
1	8694.286	75590609.050	59.461	3535.611	3105.101	9641652.220	4.031	16.249	2.800	0.019	2.863	2.800	0.019
2	6026.705	36321173.157	19.154	366.876	2263.265	5122368.460	6.841	46.799	2.663	0.012	2.738	2.663	0.012
4	2693.170	7253164.649	7.088	50.240	1100.814	1211791.463	6.487	42.081	2.447	0.016	2.518	2.447	0.016
6	1865.266	3479217.251	8.983	80.694	820.247	672805.141	4.864	23.658	2.274	0.017	2.332	2.274	0.017
8	1420.884	2018911.341	10.458	109.370	655.003	429028.930	6.628	43.930	2.169	0.027	2.172	2.169	0.027
12	999.424	998848.332	6.823	46.553	483.191	233473.542	6.717	45.118	2.068	0.032	1.910	2.068	0.032
16	771.131	594643.019	5.542	30.714	395.352	156303.204	5.390	29.052	1.950	0.030	1.706	1.950	0.030
n=1792, C=1					n=1792,C=3				Results				
p	$\overline{x}$	$\left(\overline{x}\right)^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$\left(\overline{y}\right)^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1792	exp_1792	$\pm\sigma_z$
1	11955.905	142943664.369	30.074	904.445	4248.906	18053202.197	7.593	57.654	2.814	0.009	2.891	2.814	0.009
2	8164.715	66662571.031	10.121	102.435	3042.575	9257262.631	6.780	45.968	2.683	0.007	2.789	2.683	0.007
4	4202.467	17660728.886	12.489	155.975	1584.232	2509791.030	8.540	72.932	2.653	0.016	2.607	2.653	0.016
6	2903.388	8429661.879	6.421	41.229	1159.776	1345080.370	5.712	32.627	2.503	0.014	2.447	2.503	0.014
8	2067.048	4272687.434	7.440	55.354	894.350	799861.923	3.619	13.097	2.311	0.013	2.306	2.311	0.013
12	1563.484	2444482.218	6.316	39.892	749.870	562305.017	4.289	18.396	2.085	0.015	2.069	2.085	0.015
16	1135.359	1289040.059	7.864	61.842	568.738	323462.913	5.913	34.964	1.996	0.025	1.877	1.996	0.025

Appendix III Experimental Data
(Case 2)

Table – A13 Empirical Grid Speedups On 2 CEs

n/p	G	StdDev	G	StdDev	G	StdDev	G	StdDev	G	StdDev	G	StdDev
32	0.421	0.011										
43	0.639	0.020										
48			0.763	0.013								
64	0.814	0.019	0.895	0.014	0.912	0.017						
80							1.121	0.026				
86	0.967	0.019			1.118	0.012						
96			1.149	0.024					1.412	0.015		
107							1.233	0.016				
112											1.568	0.024
128	1.084	0.017	1.279	0.033	1.324	0.020			1.508	0.017		
150											1.621	0.019
160							1.391	0.010				
171					1.451	0.018						
192			1.455	0.013					1.734	0.019		
214							1.466	0.007				
224											1.774	0.020
256	1.391	0.012			1.626	0.015			1.790	0.013		
299											1.748	0.017
320							1.572	0.018				
384			1.677	0.012					1.821	0.006		
448											1.783	0.011
512	1.707	0.012			1.777	0.010						
640							1.711	0.006				
768			1.870	0.007					1.935	0.008		
896											1.851	0.009
1024					1.904	0.009						
1280							1.789	0.004				
1536									1.976	0.010		
1792											1.926	0.016

Table – A14 Empirical Grid Speedups On 3 CEs

n/p	G	StdDev	G	StdDev	G	StdDev	G	StdDev	G	StdDev	G	StdDev
32	0.372	0.013										
43	0.670	0.034										
48			0.823	0.037								
64	0.877	0.031	1.020	0.050	1.116	0.027						
80							1.314	0.015				
86	1.129	0.030			1.291	0.036						
96			1.198	0.027					1.532	0.027		
107							1.683	0.024				
112											1.690	0.031
128	1.286	0.037	1.426	0.028	1.558	0.031			1.826	0.038		
150											1.933	0.018
160							1.882	0.018				
171					2.002	0.039						
192			1.796	0.023					2.162	0.022		
214							2.161	0.041				
224											2.330	0.024
256	1.803	0.039			2.171	0.034			2.331	0.019		
299											2.410	0.035
320							2.248	0.034				
384			2.422	0.050					2.607	0.023		
448											2.625	0.038
512	2.021	0.030			2.430	0.038						
640							2.547	0.015				
768			2.588	0.017					2.721	0.012		
896											2.887	0.014
1024					2.618	0.012						
1280							2.603	0.009				
1536									2.895	0.014		
1792											2.945	0.042

Table-A15 Empirical Grid efficiencies on 2 CEs

n/p	γ	±Std Dev	γ	±Std Dev	γ	±StdDev	γ	±Std Dev	γ	±Std Dev	γ	±StdDev
32	0.211	0.006										
43	0.320	0.010										
48			0.382	0.007								
64	0.407	0.010	0.448	0.007	0.456	0.009						
80							0.561	0.013				
86	0.484	0.010			0.559	0.006						
96			0.575	0.012					0.706	0.008		
107							0.616	0.008				
112											0.784	0.012
128	0.542	0.009	0.640	0.017	0.662	0.010			0.754	0.009		
150											0.811	0.009
160							0.696	0.005				
171					0.726	0.009						
192			0.728	0.007					0.867	0.010		
214							0.733	0.004				
224											0.887	0.010
256	0.696	0.006			0.813	0.007			0.895	0.007		
299											0.874	0.008
320							0.786	0.009				
384			0.839	0.006					0.911	0.003		
448											0.892	0.005
512	0.854	0.006			0.889	0.005						
640							0.855	0.003				
768			0.935	0.003					0.968	0.004		
896											0.926	0.005
1024					0.952	0.005						
1280							0.894	0.002				
1536									0.988	0.005		
1792											0.963	0.008

Table-A16 Empirical Grid efficiencies on 3 CEs

n/p	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev	γ	±StdDev
32	0.124	0.004										
43	0.223	0.011										
48			0.274	0.012								
64	0.292	0.010	0.340	0.017	0.372	0.009						
80							0.438	0.005				
86	0.376	0.010			0.430	0.012						
96			0.399	0.009					0.511	0.009		
107							0.561	0.008				
112											0.563	0.010
128	0.429	0.012	0.475	0.009	0.519	0.010			0.609	0.013		
150											0.644	0.006
160							0.627	0.006				
171					0.667	0.013						
192			0.599	0.008					0.721	0.007		
214							0.720	0.014				
224											0.777	0.008
256	0.601	0.013			0.724	0.011			0.777	0.006		
299											0.803	0.012
320							0.749	0.011				
384			0.807	0.017					0.869	0.008		
448											0.875	0.013
512	0.674	0.010			0.810	0.013						
640							0.849	0.005				
768			0.863	0.006					0.907	0.004		
896											0.962	0.005
1024					0.873	0.004						
1280							0.868	0.003				
1536									0.965	0.005		
1792											0.982	0.014

Table – A17 Experimental Data on 2 CEs

(x, y in seconds, Sample=10)

n=512, C=1					n=512,C=2				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_512	exp_512	$\pm\sigma_z$
1	933.738	871866.653	2.905	8.439	547.016	299226.504	3.469	12.034	1.707	0.012	1.785	1.707	0.012
2	493.533	243574.822	1.617	2.615	354.889	125946.202	2.899	8.404	1.391	0.012	1.612	1.391	0.012
4	255.688	65376.353	1.558	2.427	235.862	55630.883	3.349	11.216	1.084	0.017	1.354	1.084	0.017
6	193.566	37467.796	0.419	0.176	200.258	40103.267	3.969	15.753	0.967	0.019	1.168	0.967	0.019
8	132.243	17488.211	1.544	2.384	162.456	26391.952	3.361	11.296	0.814	0.019	1.028	0.814	0.019
12	97.272	9461.842	1.190	1.416	152.229	23173.668	4.507	20.313	0.639	0.020	0.832	0.639	0.02
16	52.741	2781.613	0.624	0.389	125.288	15697.083	3.465	12.006	0.421	0.013	0.701	0.421	0.011
n=768, C=1					n=768,C=2				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_768	exp_768	$\pm\sigma_z$
1	2143.256	4593546.282	2.810	7.896	1146.358	1314136.664	3.734	13.943	1.870	0.007	1.883	1.870	0.007
2	1147.505	1316767.725	1.963	3.853	684.451	468473.171	4.654	21.660	1.677	0.012	1.780	1.677	0.012
4	565.776	320102.482	1.230	1.513	388.975	151301.551	3.388	11.479	1.455	0.013	1.605	1.455	0.013
6	350.044	122530.802	5.555	30.858	273.657	74888.154	5.607	31.438	1.279	0.033	1.462	1.279	0.033
8	284.629	81013.668	3.029	9.175	247.780	61394.928	4.352	18.940	1.149	0.024	1.343	1.149	0.024
12	203.709	41497.357	0.759	0.576	227.647	51823.157	3.562	12.688	0.895	0.014	1.156	0.895	0.014
16	158.084	24990.551	0.619	0.383	207.276	42963.340	3.520	12.390	0.763	0.013	1.017	0.763	0.013
n=1024, C=1					n=1024,C=2				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1024	exp_1024	$\pm\sigma_z$
1	3814.977	14554049.511	11.799	139.216	2003.331	4013335.096	7.648	58.492	1.904	0.009	1.924	1.904	0.009
2	1863.004	3470783.904	4.722	22.297	1048.612	1099587.127	5.482	30.052	1.777	0.010	1.853	1.777	0.010
4	970.856	942561.373	5.519	30.459	597.102	356530.798	4.283	18.344	1.626	0.015	1.727	1.626	0.015
6	718.574	516348.593	2.819	7.947	495.245	245267.610	5.990	35.880	1.451	0.018	1.618	1.451	0.018
8	505.028	255053.281	1.078	1.162	381.560	145588.034	5.634	31.742	1.324	0.020	1.522	1.324	0.020
12	360.360	129859.330	0.500	0.250	322.349	103908.878	3.441	11.840	1.118	0.012	1.362	1.118	0.012
16	262.711	69017.070	0.694	0.482	288.020	82955.520	5.388	29.031	0.912	0.017	1.233	0.912	0.017

(Continue…)

n=1280, C=1					n=1280,C=2				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1280	exp_1280	$\pm\sigma_z$
1	5170.966	26738889.373	6.296	39.640	2890.528	8355152.119	5.383	28.977	1.789	0.004	1.945	1.789	0.004
2	2827.949	7997295.547	4.875	23.766	1652.904	2732091.633	5.026	25.261	1.711	0.006	1.893	1.711	0.006
4	1413.488	1997948.326	7.216	52.071	899.434	808981.520	9.192	84.493	1.572	0.018	1.797	1.572	0.018
6	1119.727	1253788.555	1.271	1.615	763.692	583225.471	3.739	13.980	1.466	0.007	1.710	1.466	0.007
8	771.394	595048.703	3.651	13.330	554.486	307454.724	3.221	10.375	1.391	0.010	1.632	1.391	0.010
12	562.241	316114.942	2.740	7.508	456.038	207970.657	5.599	31.349	1.233	0.016	1.496	1.233	0.016
16	405.114	164117.353	1.720	2.958	361.242	130495.783	8.304	68.956	1.121	0.026	1.381	1.121	0.026
n=1536, C=1					n=1536,C=2				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1536	exp_1536	$\pm\sigma_z$
1	8797.210	77390903.784	26.915	724.417	4452.973	19828968.539	18.876	356.303	1.976	0.010	1.957	1.976	0.010
2	4564.836	20837727.707	8.052	64.835	2358.740	5563654.388	8.687	75.464	1.935	0.008	1.916	1.935	0.008
4	2297.847	5280100.835	1.983	3.932	1261.731	1591965.116	4.093	16.753	1.821	0.006	1.840	1.821	0.006
6	1543.936	2383738.372	0.594	0.353	862.693	744239.212	6.365	40.513	1.790	0.013	1.769	1.790	0.013
8	1161.329	1348685.046	1.468	2.155	669.728	448535.594	7.338	53.846	1.734	0.019	1.704	1.734	0.019
12	763.046	582239.198	1.418	2.011	506.045	256081.542	5.703	32.524	1.508	0.017	1.588	1.508	0.017
16	586.456	343930.640	0.814	0.663	415.277	172454.987	4.419	19.528	1.412	0.015	1.487	1.412	0.015
n=1792, C=1					n=1792,C=2				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1792	exp_1792	$\pm\sigma_z$
1	11290.790	127481938.824	30.106	906.371	5861.431	34356373.368	46.121	2127.147	1.926	0.016	1.965	1.926	0.016
2	5735.851	32899986.694	18.685	349.129	3098.701	9601947.887	11.880	141.134	1.851	0.009	1.932	1.851	0.009
4	2944.210	8668372.524	4.354	18.957	1651.725	2728195.476	9.850	97.023	1.783	0.011	1.869	1.783	0.011
6	1865.064	3478463.724	5.038	25.381	1067.202	1138920.109	9.681	93.722	1.748	0.017	1.810	1.748	0.017
8	1468.701	2157082.627	4.883	23.844	827.821	685287.608	9.009	81.162	1.774	0.020	1.754	1.774	0.020
12	1034.723	1070651.687	2.029	4.117	638.448	407615.849	7.286	53.086	1.621	0.019	1.654	1.621	0.019
16	773.222	597872.261	5.152	26.543	493.065	243113.094	6.909	47.734	1.568	0.024	1.564	1.568	0.024

Table – A18 Experimental Data On 3 CEs

n=512, C=1					n=512,C=3				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_512	exp_512	$\pm\sigma_z$
1	933.738	871866.653	2.905	8.439	461.999	213443.076	6.612	43.719	2.021	0.030	2.296	2.021	0.030
2	493.533	243574.822	1.617	2.615	273.732	74929.208	5.890	34.692	1.803	0.039	1.861	1.803	0.039
4	255.688	65376.353	1.558	2.427	198.796	39519.850	5.599	31.349	1.286	0.037	1.355	1.286	0.037
6	193.566	37467.796	0.419	0.176	171.393	29375.560	4.587	21.041	1.129	0.030	1.067	1.129	0.030
8	132.243	17488.211	1.544	2.384	150.781	22734.910	4.970	24.701	0.877	0.031	0.881	0.877	0.031
12	97.272	9461.842	1.190	1.416	145.255	21099.015	7.267	52.809	0.670	0.034	0.656	0.670	0.034
16	52.741	2781.613	0.624	0.389	141.705	20080.307	4.506	20.304	0.372	0.013	0.525	0.372	0.013
n=768, C=1					n=768,C=3				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_768	exp_768	$\pm\sigma_z$
1	2143.256	4593546.282	2.810	7.896	828.181	685883.769	5.406	29.225	2.588	0.017	2.608	2.588	0.017
2	1147.505	1316767.725	1.963	3.853	473.755	224443.800	9.675	93.606	2.422	0.050	2.308	2.422	0.050
4	565.776	320102.482	1.230	1.513	315.105	99291.161	4.058	16.467	1.796	0.023	1.879	1.796	0.023
6	350.044	122530.802	5.555	30.858	245.367	60204.965	2.906	8.445	1.427	0.028	1.585	1.426	0.028
8	284.629	81013.668	3.029	9.175	237.623	56464.690	4.829	23.319	1.198	0.027	1.372	1.198	0.027
12	203.709	41497.357	0.759	0.576	199.687	39874.898	9.853	97.082	1.020	0.050	1.084	1.020	0.050
16	158.084	24990.551	0.619	0.383	192.055	36885.123	8.501	72.267	0.823	0.037	0.897	0.823	0.037
n=1024, C=1					n=1024,C=3				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1024	exp_1024	$\pm\sigma_z$
1	3814.977	14554049.511	11.799	139.216	1457.384	2123968.123	4.641	21.539	2.618	0.012	2.747	2.618	0.012
2	1863.004	3470783.904	4.722	22.297	766.627	587716.957	11.729	137.569	2.430	0.038	2.533	2.430	0.038
4	970.856	942561.373	5.519	30.459	447.289	200067.450	6.530	42.641	2.171	0.034	2.194	2.171	0.034
6	718.574	516348.593	2.819	7.947	358.909	128815.670	6.871	47.211	2.002	0.039	1.936	2.002	0.039
8	505.028	255053.281	1.078	1.162	324.159	105079.057	6.375	40.641	1.558	0.031	1.733	1.558	0.031
12	360.360	129859.330	0.500	0.250	279.039	77862.764	7.865	61.858	1.291	0.036	1.434	1.291	0.036
16	262.711	69017.070	0.694	0.482	235.365	55396.683	5.646	31.877	1.116	0.027	1.224	1.116	0.027

(Continue…)

n=1280, C=1					n=1280,C=3				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1280	exp_1280	$\pm\sigma_z$
1	5170.966	26738889.373	6.296	39.640	1986.773	3947266.954	6.228	38.788	2.603	0.009	2.819	2.603	0.009
2	2827.949	7997295.547	4.875	23.766	1110.363	1232905.992	6.360	40.450	2.547	0.015	2.660	2.547	0.015
4	1413.488	1997948.326	7.216	52.071	628.637	395184.478	8.995	80.910	2.248	0.034	2.390	2.248	0.034
6	1119.727	1253788.555	1.271	1.615	518.153	268482.531	9.786	95.766	2.161	0.041	2.170	2.161	0.041
8	771.394	595048.703	3.651	13.330	409.909	168025.388	3.517	12.369	1.882	0.018	1.988	1.882	0.018
12	562.241	316114.942	2.740	7.508	334.111	111630.160	4.511	20.349	1.683	0.024	1.704	1.683	0.024
16	405.114	164117.353	1.720	2.958	308.406	95114.261	3.325	11.056	1.314	0.015	1.492	1.314	0.015
n=1536, C=1					n=1536,C=3				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1536	exp_1536	$\pm\sigma_z$
1	8797.210	77390903.784	26.915	724.417	3038.425	9232026.481	10.934	119.552	2.895	0.014	2.863	2.895	0.014
2	4564.836	20837727.707	8.052	64.835	1677.596	2814328.339	6.452	41.628	2.721	0.012	2.738	2.721	0.012
4	2297.847	5280100.835	1.983	3.932	881.257	776613.900	7.660	58.676	2.607	0.023	2.518	2.607	0.023
6	1543.936	2383738.372	0.594	0.353	662.239	438560.493	5.315	28.249	2.331	0.019	2.332	2.331	0.019
8	1161.329	1348685.046	1.468	2.155	537.163	288544.089	5.493	30.173	2.162	0.022	2.172	2.162	0.022
12	763.046	582239.198	1.418	2.011	417.944	174677.187	8.687	75.464	1.826	0.038	1.910	1.826	0.038
16	586.456	343930.640	0.814	0.663	382.911	146620.834	6.665	44.422	1.532	0.027	1.706	1.532	0.027
n=1792, C=1					n=1792,C=3				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1792	exp_1792	$\pm\sigma_z$
1	11290.790	127481938.824	30.106	906.371	3833.679	14697094.675	54.104	2927.243	2.945	0.042	2.891	2.945	0.042
2	5735.851	32899986.694	18.685	349.129	1986.815	3947433.844	7.064	49.900	2.887	0.014	2.789	2.887	0.014
4	2944.210	8668372.524	4.354	18.957	1121.473	1257701.690	16.237	263.640	2.625	0.038	2.607	2.625	0.038
6	1865.064	3478463.724	5.038	25.381	773.993	599065.164	11.035	121.771	2.410	0.035	2.447	2.410	0.035
8	1468.701	2157082.627	4.883	23.844	630.430	397441.985	6.273	39.351	2.330	0.024	2.306	2.330	0.024
12	1034.723	1070651.687	2.029	4.117	535.254	286496.845	4.915	24.157	1.933	0.018	2.069	1.933	0.018
16	773.222	597872.261	5.152	26.543	457.659	209451.760	7.943	63.091	1.690	0.031	1.877	1.690	0.031

Appendix IV Experimental Data

(Case 3)

Table – A19 Empirical Grid Speedups on 4CEs

n/p	G	StdDev	G	StdDev	G	StdDev	G	StdDev	G	StdDev	G	StdDev
32	0.185	±0.006										
48			0.653	±0.015								
64					0.919	±0.031						
80							1.111	±0.018				
96									1.474	±0.019		
112											1.788	±0.036
128	1.415	±0.020										
192			2.159	±0.042								
256					2.580	±0.044						
320							2.942	±0.035				
384									3.352	±0.029		
448											3.154	±0.050
512	2.359	±0.038										
768			3.040	±0.039								
1024					3.200	±0.040						
1280							3.587	±0.042				
1536									3.745	±0.057		
1792											3.515	±0.038

Table – A20 Empirical *Grid Efficiencies on* 4 *CEs*

n/p	γ	StdDev	γ	StdDev	γ	StdDev	γ	StdDev	γ	StdDev	γ	StdDev
32	0.046	±0.001										
48			0.163	±0.004								
64					0.230	±0.008						
80							0.278	±0.004				
96									0.369	±0.005		
112											0.447	±0.009
128	0.354	±0.005										
192			0.540	±0.011								
256					0.645	±0.011						
320							0.736	±0.009				
384									0.838	±0.007		
448											0.789	±0.013
512	0.590	±0.009										
768			0.760	±0.010								
1024					0.800	±0.010						
1280							0.897	±0.010				
1536									0.936	±0.014		
1792											0.879	±0.010

Table – A21 *Experimental Data on 4 CEs*

n−512,C−1					n−512,C−4				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_512	exp_512	$\pm\sigma_z$
1	916.210	839440.764	8.604	74.029	388.328	150798.636	5.071	25.715	2.359	0.038	2.856	2.359	0.038
4	257.275	66190.426	0.324	0.105	181.823	33059.603	2.606	6.791	1.415	0.020	1.631	1.415	0.020
16	46.327	2146.191	0.328	0.108	250.733	62867.037	7.521	56.565	0.185	0.006	0.340	0.185	0.006
n−768,C−1					n−768,C−4				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_768	exp_768	$\pm\sigma_z$
1	2153.101	4635843.916	13.476	181.603	708.290	501674.724	7.937	62.996	3.040	0.039	3.348	3.040	0.039
4	578.785	334992.076	0.787	0.619	268.125	71891.016	5.245	27.510	2.159	0.042	2.379	2.159	0.042
16	154.973	24016.631	1.430	2.045	237.453	56383.927	5.137	26.389	0.653	0.015	0.670	0.653	0.015
n−1024,C−1					n−1024,C−4				Results				
p	$\overline{x}$	$(\overline{x})^2$	$\pm\sigma_x$	σ_x^2	$\overline{y}$	$(\overline{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1024	exp_1024	$\pm\sigma_z$
1	3326.445	11065236.338	22.495	506.025	1042.916	1087673.783	10.857	117.874	3.190	0.040	3.575	3.200	0.040
4	972.210	945192.284	7.780	60.528	376.841	142009.139	5.628	31.674	2.580	0.044	2.851	2.580	0.044
16	267.865	71751.658	2.548	6.492	291.576	85016.564	9.422	88.774	0.919	0.031	1.029	0.919	0.031
n−1280,C−1					n−1280,C−4				Results				

p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1280	exp_1280	$\pm\sigma_z$
1	6023.714	36285130.354	61.126	3736.388	1679.465	2820602.686	9.574	91.661	3.587	0.042	3.697	3.587	0.042
4	1670.047	2789056.982	1.063	1.130	567.655	322232.199	6.772	45.860	2.942	0.035	3.149	2.942	0.035
16	424.594	180280.065	1.655	2.739	382.204	146079.898	5.842	34.129	1.111	0.018	1.378	1.111	0.018
n−1536,C−1					n−1536,C−4				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1536	exp_1536	$\pm\sigma_z$
1	8505.125	72337151.266	47.148	2222.934	2270.777	5156428.184	32.228	1038.644	3.745	0.057	3.769	3.745	0.057
4	2403.610	5777341.032	1.608	2.586	717.024	514123.417	6.279	39.426	3.352	0.029	3.344	3.352	0.029
16	618.205	382177.422	1.208	1.459	419.325	175833.456	5.207	27.113	1.474	0.019	1.697	1.474	0.019
n−1792,C−1					n−1792,C−4				Results				
p	$\bar{x}$	$(\bar{x})^2$	$\pm\sigma_x$	σ_x^2	$\bar{y}$	$(\bar{y})^2$	$\pm\sigma_y$	σ_y^2	G	±StdDev	ana_1792	exp_1792	$\pm\sigma_z$
1	9972.208	99444932.395	72.974	5325.205	2837.289	8050208.870	22.861	522.625	3.515	0.038	3.817	3.515	0.038
4	3092.117	9561187.542	37.435	1401.379	980.233	960856.734	10.238	104.817	3.154	0.050	3.477	3.154	0.050
16	861.687	742504.486	13.767	189.530	481.943	232269.055	5.869	34.445	1.788	0.036	1.977	1.788	0.036

References:

[Hoekstra05] Alfons Hoekstra etc, " Grid Speedup and Scalability of Tightly Coupled Parallel Applications in a Grid Computing Environment".

[Kumar87] Kumar, V. and Rao, V. N., "Parallel depth first search. part II. analysis," International Journal of Parallel Programming, vol. 16, no. 6, pp. 501-519, 1987

[Eager89] Eager, D. L., Zahorjan, J. and Lazowska, E. D., "Speedup versus efficiency in parallel systems,"IEEE Transactions on Computers, vol. 38, no. 3, pp. 408-423, March 1989.

[Hwang93] Kai Hwang: "Advanced Computer Architecture: Parallelism, Scalability, Programmability". MacGraw-Hill, Inc. ISBN 0-07-113342-9. 1993.

[Gustafson88] Gustafson, J. L., "Reevaluating Amdahl's law," Communications of the ACM, vol. 31, no.5, pp. 532-533, May 1988.

[Gustafson88a] Gustafson, J. L., Montry, G. R. and Benner, R. E., "Development of parallel methods for a 1024-processor hypercube," SIAM Journal on Scientific and Statistical Computing, vol. 9, no. 4, pp.609-638, 1988.

[DAS-2web] DAS-2 Supercomputer available http://www.cs.vu.nl/das2/

[Myrinetweb] Myrinet company available http://www.myricom.com

[Hockner94] Cliff Addison, Vladimir Getov, Anthony J. G. Hey, Roger W. Hockney, I. C. Wolton: Benchmarking for Distributed Memory Parallel Systems: Gaining Insight from Numbers. 1653-1668. Parallel Computing, Volume 20, Numbers 10-11, 3 November 1994

[Culler93] "LogP:Towards a Realistic Model of Parallel Computation", D. E. Culler, R. M. Karp, D. A. Patterson, A. Sahay, K. E. Schauser, E. Santos, R. Subramonian, T. von Eicken, , 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, San Diego, CA, May 1993.

[Alexandrov95] "LogGP: Incorporating Long Messages into the LogP Model --- One step closer towards a realistic model for parallel computation", Albert Alexandrov, Mihai F. Ionescu, Klaus E. Schauser, Chris Scheiman, TRCS95-09, Computer Science Department, University of California, Santa Barbara, April 1995.

.