Fundamentals of Grid Computing

Burhan Eyimaya
200311021

Motivation

- Grid computing, most simply stated, is distributed computing taken to the next evolutionary level.
- The goal is to create the illusion of a simple yet large and powerful self managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources.
- The following major topics will be introduced to the readers:
  - � What grid computing can do
  - � Grid concepts and components
  - � What the grid cannot do
  - � The present and the future

Exploiting underutilized resources

- In most organizations, there are large amounts of underutilized computing resources.
- Most desktop machines are busy less than 5% of the time.
- In some organizations, even the server machines can often be relatively idle.
- Grid computing provides a framework for exploiting these underutilized resources and thus has the possibility of substantially increasing the efficiency of resource usage.

Parallel CPU capacity

- The potential for massive parallel CPU capacity is one of the most attractive features of a grid.
- In addition to pure scientific needs, such computing power is driving a new evolution in industries such as the bio-medical field, financial modeling, oil exploration, motion picture animation, and many others.
- The common attribute among such uses is that the applications have been written to use algorithms that can be partitioned into independently running parts. To the extent that these subjobs do not need to communicate with each other, the more “scalable” the application becomes.

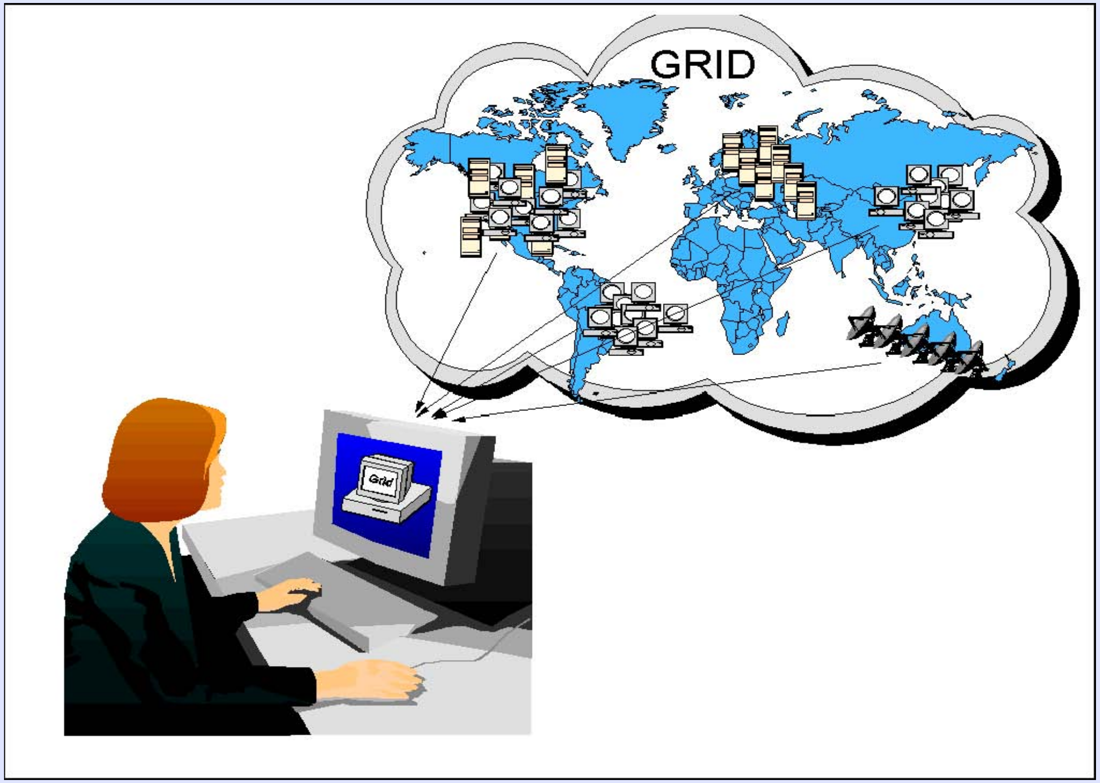
Applications

- There are many factors to consider in grid-enabling an application.
- One must understand that not all applications can be transformed to run in parallel on a grid and achieve scalability.
- There are some practical tools that skilled application designers can use to write a parallel grid application. However, automatic transformation of applications is a science in its infancy.
- This can be a difficult job and often requires top mathematics and programming talents, if it is even possible in a given situation.

Virtual resources and virtual organizations for collaboration

- Another important grid computing contribution is to enable and simplify collaboration among a wider audience.
- In the past, distributed computing promised this collaboration and achieved it to some extent.
- Grid computing takes these capabilities to an even wider audience, while offering important standards that enable very heterogeneous systems to work together to form the image of a large virtual computing system offering a variety of virtual resources, as illustrated in Figure 1.

Figure 1



The Grid virtualizes heterogeneous and geographically disperse resources for each virtual organization presenting a simpler view

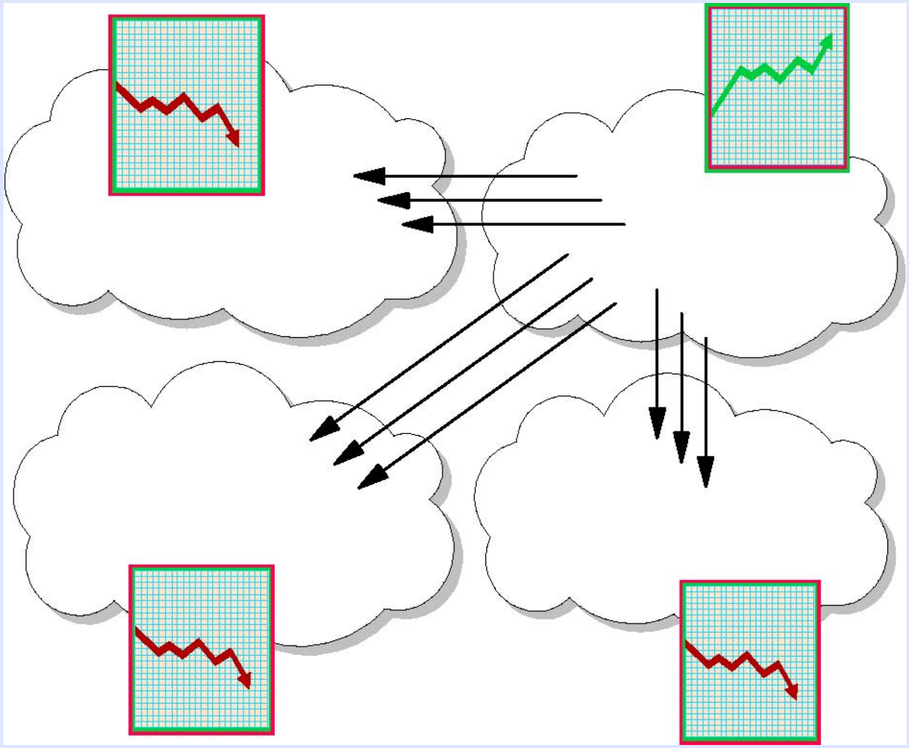
Access to additional resources

- Some machines on the grid may have special devices. Most of us have used remote printers, perhaps with advanced color capabilities or faster speeds.
- Similarly, a grid can be used to make use of other special equipment.
- For example, a machine may have a high speed, self feeding, DVD writer that could be used to publish a quantity of data faster. Some machines on the grid may be connected to scanning electron microscopes that can be operated remotely.

Resource balancing

- For applications that are grid enabled, the grid can offer a resource balancing effect by scheduling grid jobs on machines with low utilization, as illustrated in Figure 2.
- This feature can prove invaluable for handling occasional peak loads of activity in parts of an larger organization. This can happen in two ways:
  - An unexpected peak can be routed to relatively idle machines in the grid.
  - If the grid is already fully utilized, the lowest priority work being performed on the grid can be temporarily suspended or even cancelled and performed again later to make room for the higher priority work.

Figure 2

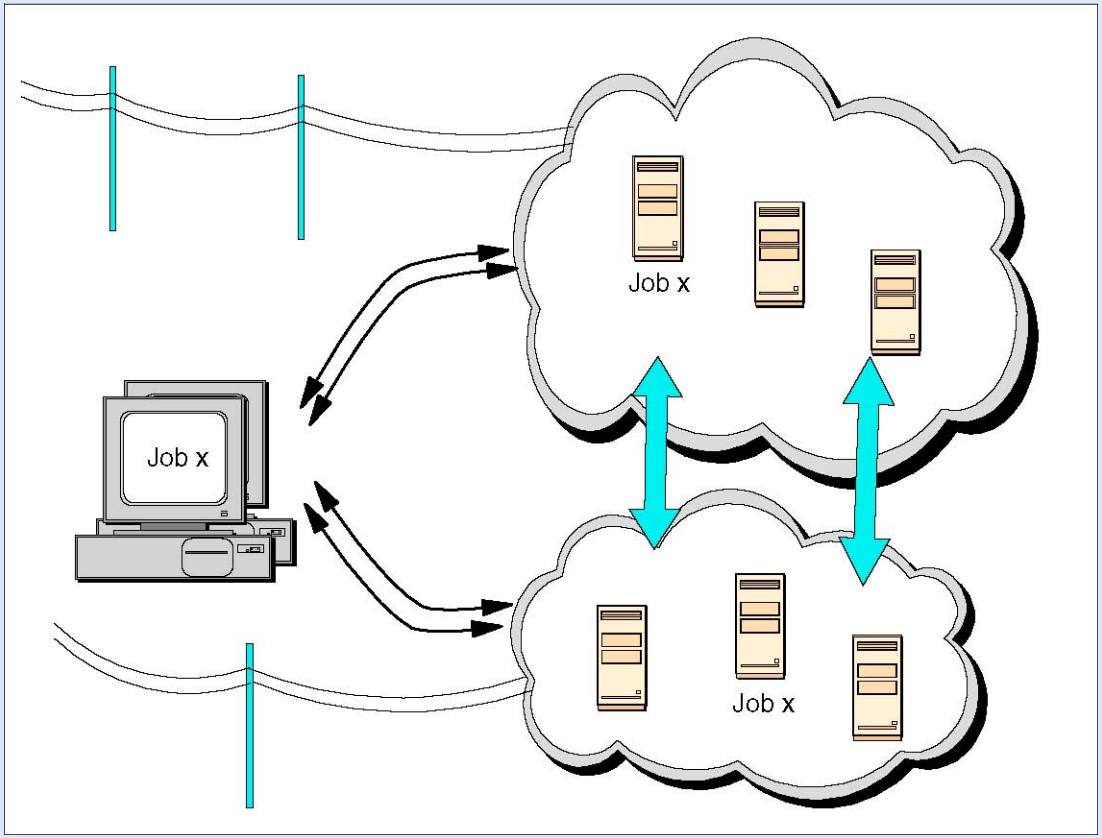


Jobs are migrated to less busy parts of the grid to balance resource loads and absorb unexpected peaks of activity in a part of an organization

Reliability

- High-end conventional computing systems use expensive hardware to increase reliability.
- They are built using chips with redundant circuits that vote on results, and contain much logic to achieve graceful recovery from an assortment of hardware failures. Power supplies and cooling systems are duplicated.
- All of this builds a reliable system, but at a great cost, due to the duplication of high-reliability components.
- In the future, we will see an alternate approach to reliability that relies more on software technology than expensive hardware. A grid is just the beginning of such technology.
- The systems in a grid can be relatively inexpensive and geographically dispersed. Thus, if there is a power or other kind of failure at one location, the other parts of the grid are not likely to be affected. Grid management software can automatically resubmit jobs to other machines on the grid when a failure is detected. This is illustrated in Figure 3

Figure 3

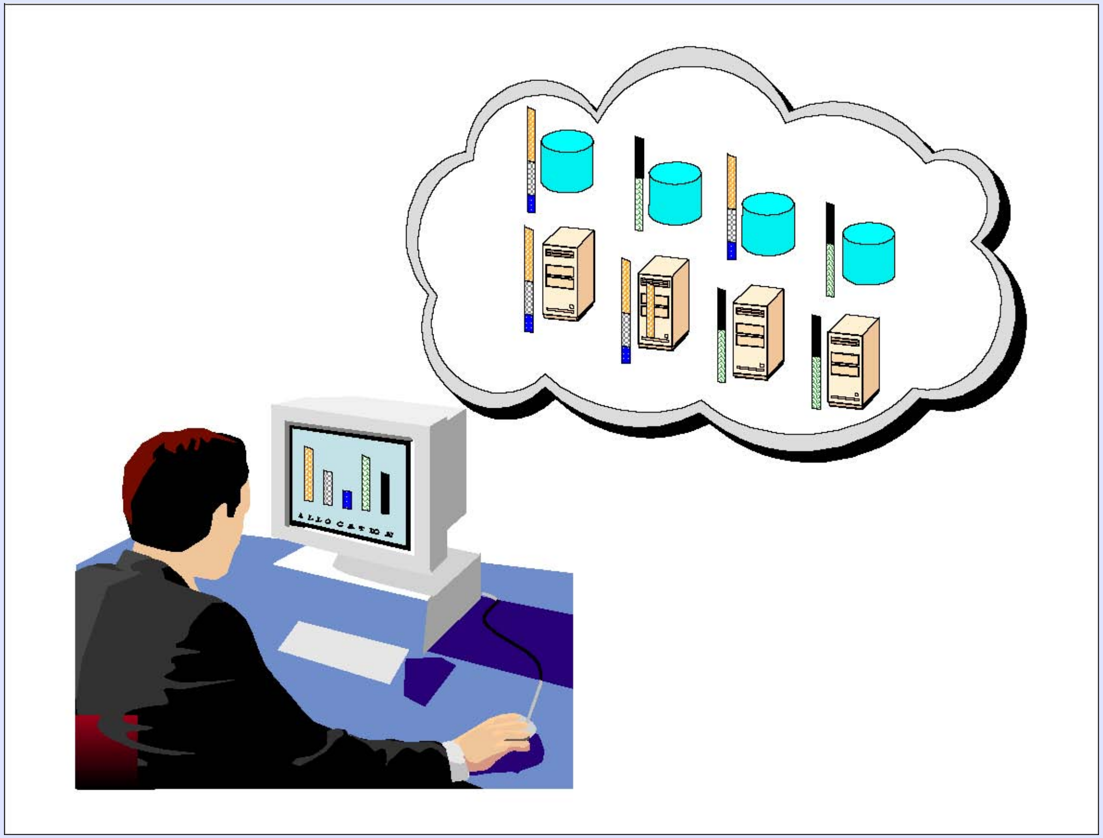


Redundant grid configuration and redundant job submission used to achieve high reliability

Management

- The grid offers management of priorities among different projects.
- In the past, each project may have been responsible for its own IT resource hardware and the expenses associated with it.
- Often this hardware might be underutilized while another project finds itself in trouble, needing more resources due to unexpected events.
- With the larger view a grid can offer, it becomes easier to control and manage such situations. As illustrated in Figure 4

Figure 4



Administrators can adjust policies to better allocate resources

Grid concepts and components

- Computation
- Storage
- Communications
- Software and licenses
- Special equipment, capacities, architectures, and policies
- Jobs and applications
- Scheduling, reservation, and scavenging

Computation

- The most common resource is computing cycles provided by the processors of the machines on the grid.
- The processors can vary in speed, architecture, software platform, memory, storage, and connectivity.
- There are three primary ways to exploit the computation resources of a grid.
- The first is to use it to run an existing application on an available machine on the grid rather than locally.
- The second is to use an application designed to split its work in a way that the separate parts can execute in parallel on different processors.
- The third is to run an application that needs to be executed many times on many different machines in the grid

Storage

- The second most common resource used in a grid is data storage.
- A grid providing an integrated view of data storage is sometimes called a “data grid.”
- Each machine on the grid usually provides some quantity of storage for grid use, even if temporary
- Capacity can be increased by using the storage on multiple machines with a unifying file system.
- Any individual file or data base can span several storage devices and machines, eliminating maximum size restrictions often imposed by file systems shipped with operating systems.

Communications

- The rapid growth in communication capacity among machines today makes grid computing practical, compared to the limited bandwidth available when distributed computing was first emerging.
- Communications within the grid are important for sending jobs and their required data to points within the grid.
- Some jobs require a large amount of data to be processed and it may not always reside on the machine running the job.
- The bandwidth available for such communications can often be a critical resource that can limit utilization of the grid.

Software and licenses

- The grid may have software installed that may be too expensive to install on every grid machine.
- Using a grid, the jobs requiring this software are sent to the particular machines on which this software happens to be installed.
- When the licensing fees are significant, this approach can save significant expenses for an organization.

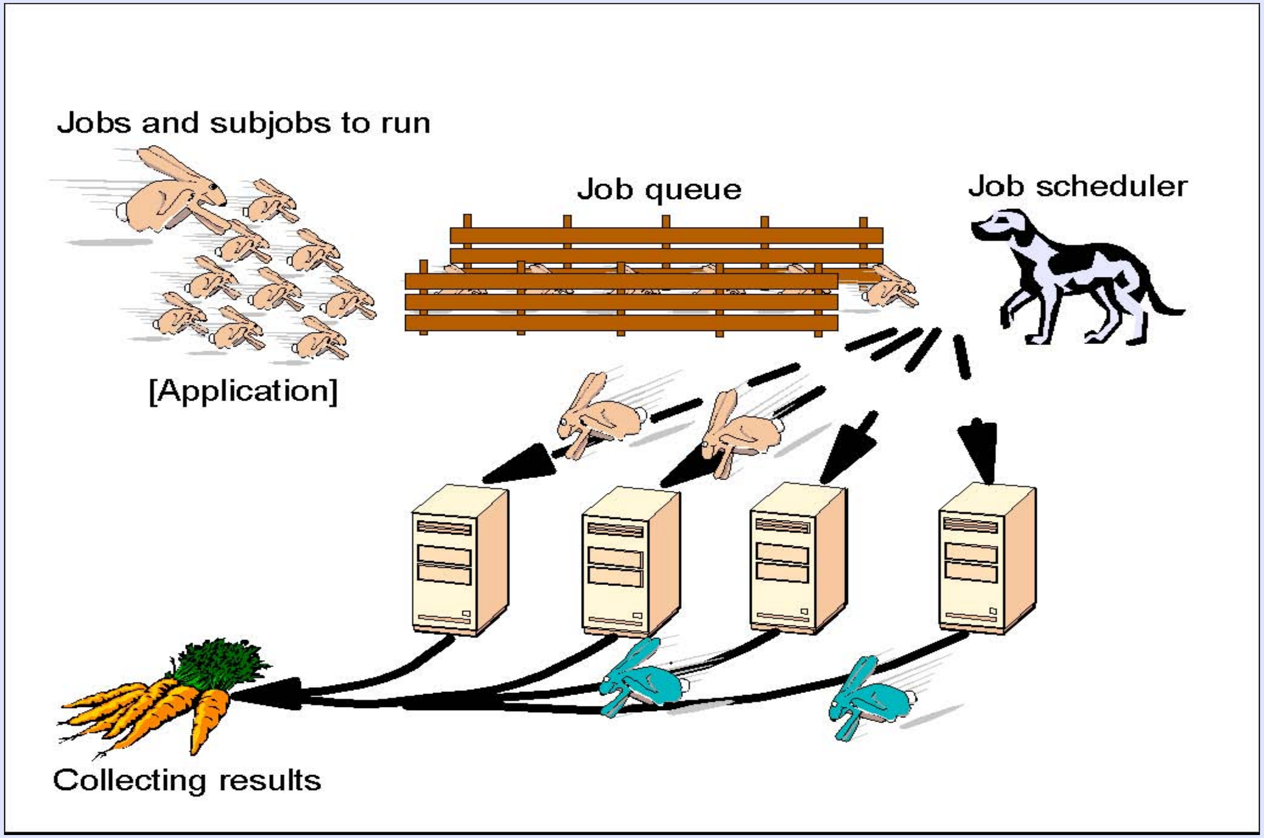
Special equipment, capacities, architectures, and policies

- Platforms on the grid will often have different architectures, operating systems, devices, capacities, and equipment.
- Each of these items represents a different kind of resource that the grid can use as criteria for assigning jobs to machines.
- While some software may be available on several architectures, for example, PowerPC and x86, such software is often designed to run only on a particular type of hardware and operating system.
- Such attributes must be considered when assigning jobs to resources in the grid.

Jobs and applications

- Jobs are programs that are executed at an appropriate point on the grid.
- They may compute something, execute one or more system commands, move or collect data, or operate machinery.
- A grid application that is organized as a collection of jobs is usually designed to have these jobs execute in parallel on different machines in the grid.
- Applications may be broken down into any number of individual jobs, as illustrated in Figure 5.

Figure 5



An application is one or more jobs that are scheduled to run on machines in the grid; the results are collected and assembled to produce the answer

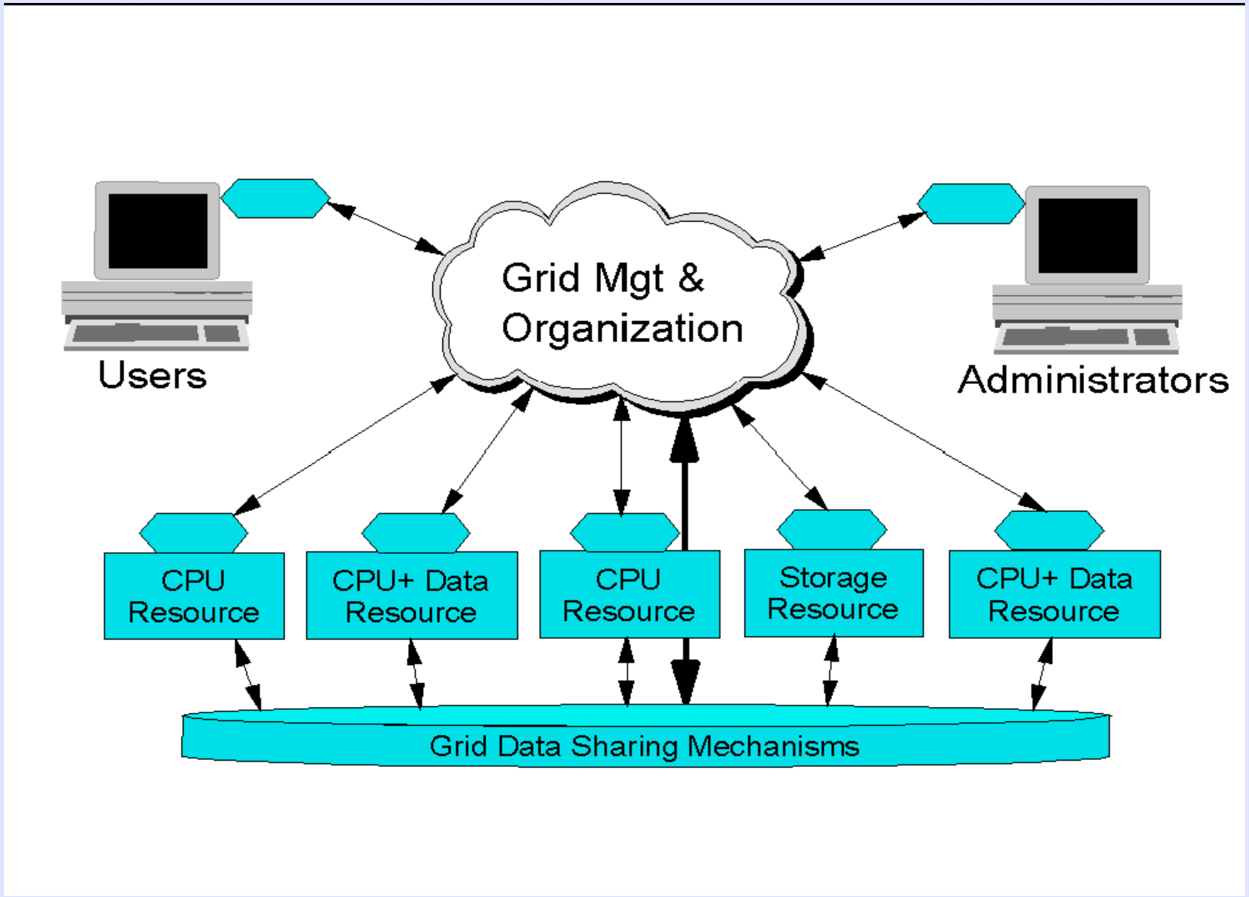
Scheduling, reservation, and scavenging

- The grid system is responsible for sending a job to a given machine to be executed.
- In the simplest of grid systems, the user may select a machine suitable for running his job and then execute a grid command that sends the job to the selected machine.
- More advanced grid systems would include a job "scheduler" of some kind that automatically finds the most appropriate machine on which to run any given job that is waiting to be executed.
- Schedulers react to current availability of resources on the grid.

Clusters

- Grids can be built in all sizes, ranging from just a few machines in a department to groups of machines organized as a hierarchy spanning the world.
- As presented in Figure 7, the simplest grid consists of just a few machines
- This kind of grid uses homogeneous systems
- The machines are usually in one department of an organization, and their use as a grid may not require any special policies or security concerns.
- Some people would call this a “cluster” implementation rather than a “grid.”

Figure 6



A simple grid

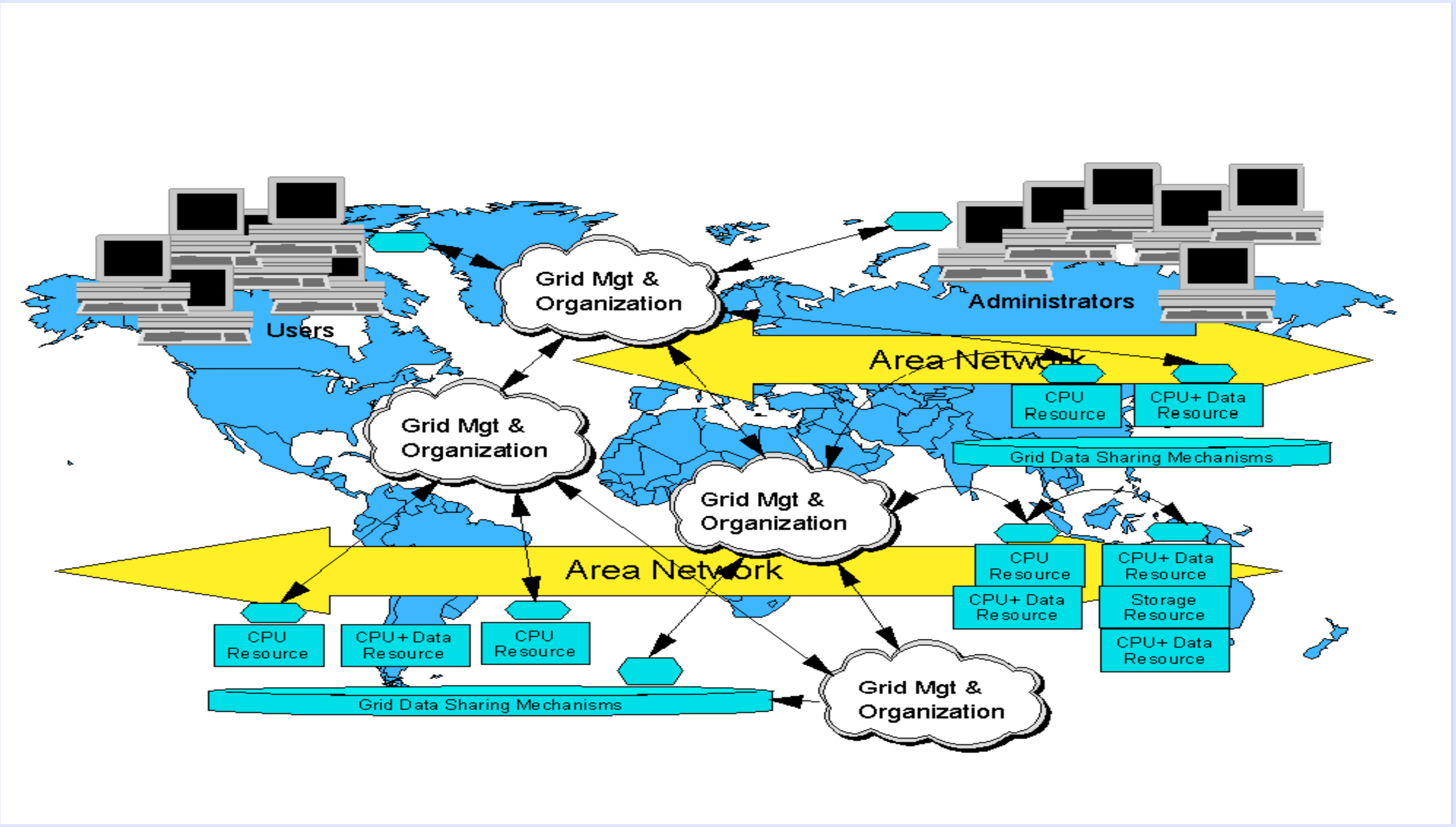
Intragrids

- The next progression would be to include heterogeneous machines.
- In this configuration, more types of resources are available.
- The grid system is likely to include some scheduling components. File sharing may still be accomplished using networked file systems.
- Machines participating in the grid may include ones from multiple departments but within the same organization. Such a grid is also referred to as an “Intragrid.”

Intergrids

- Over time, as illustrated in Figure 7, a grid may grow to cross organization boundaries, and may be used to collaborate on projects of common interest.
- This is known as an “Intergrid.” The highest levels of security are usually required in this configuration to prevent possible attacks and spying.
- The Intergrid offers the prospect for trading or brokering resources over a much wider audience.

Figure 7



A more complex Intergrid

What the grid cannot do

- The grid is not a silver bullet that can take any application and run it a 1000 times faster without the need for buying any more machines or software.
- Not every application is suitable or enabled for running on a grid. Some kinds of applications simply cannot be parallelized.
- For others, it can take a large amount of work to modify them to achieve faster throughput. The configuration of a grid can greatly affect the performance, reliability, and security of an organization’s computing infrastructure.
- For all of these reasons, it is important for the us to understand how far the grid has evolved today and which features are coming tomorrow or in the distant future.

The present and the future

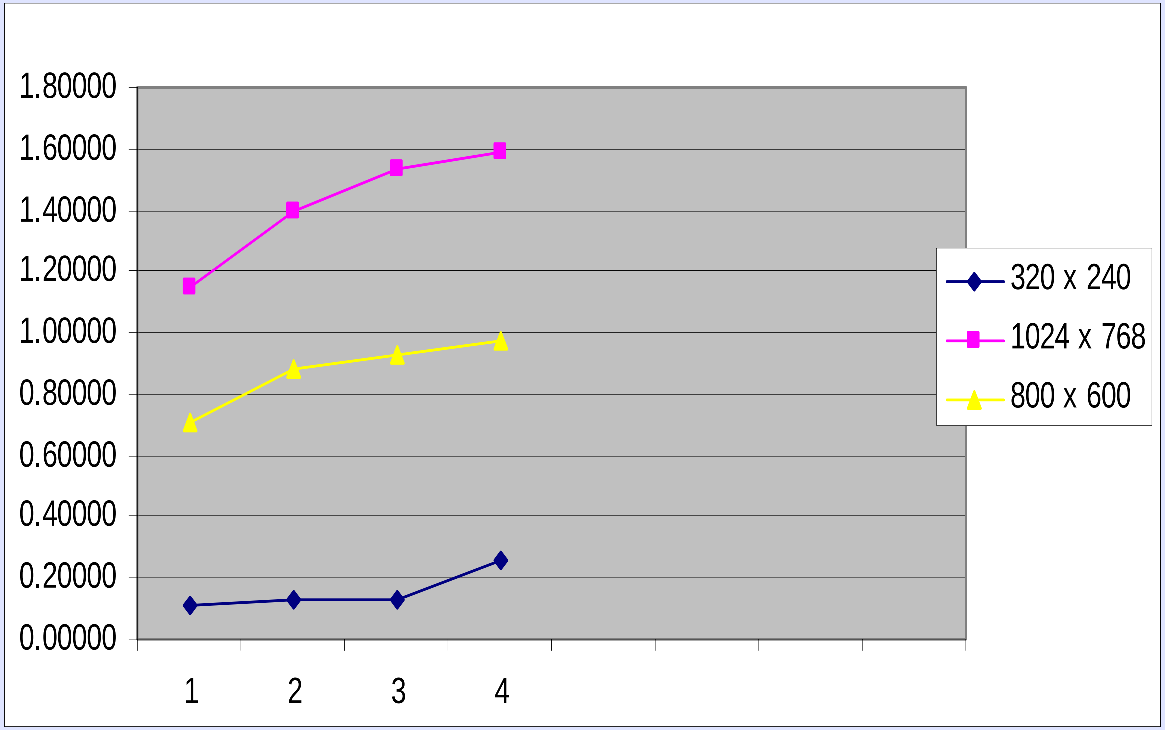
- Today, grid systems are still at the early stages of providing a reliable, well performing, and automatically recoverable virtual data sharing and storage.
- We will see products that take on this task in a grid setting, federating data of all kinds, and achieving better performance, integration with scheduling, reliability, and capacity.
- As Grids gain momentum and are more widely accepted across multiple vertical market segments worldwide it will be a very feasible solution to big projects that need huge amount of resources.

Edge Detection Time Analysis Tables

Num of Procs	320 x 240	800 x 600	1024 x 768
1	0.11078	0.71070	1.14766
2	0.13039	0.88405	1.39597
3	0.13108	0.93143	1.52944
4	0.25948	0.97370	1.58707

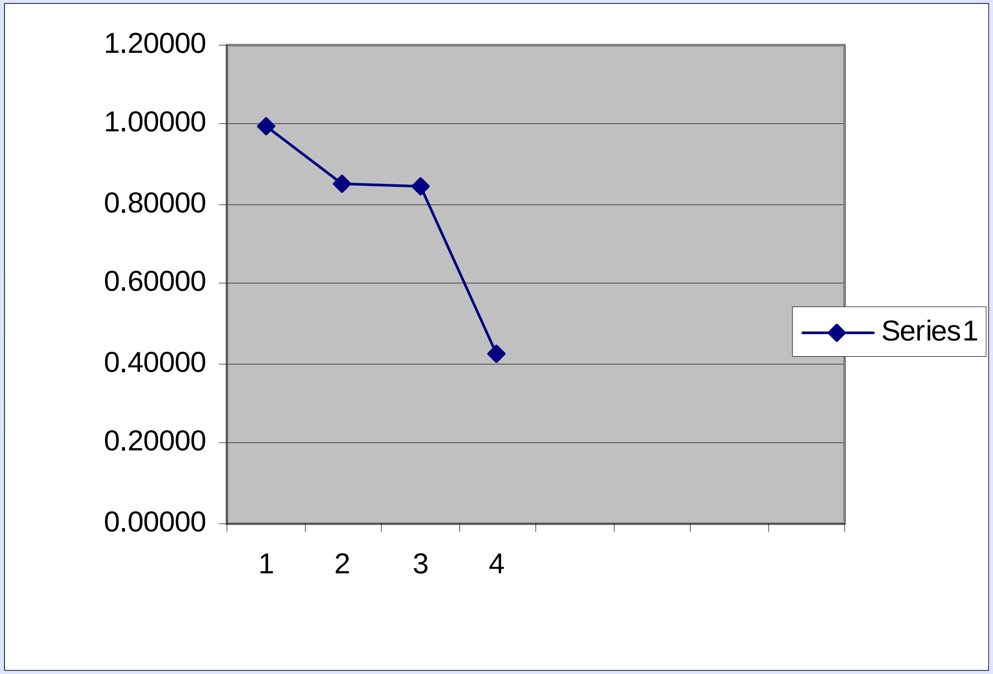
speed up	efficency
1.00000	1
0.849579716	0.424789858
0.845094599	0.2816982
0.426904934	0.106726234

CPU-Time Chart



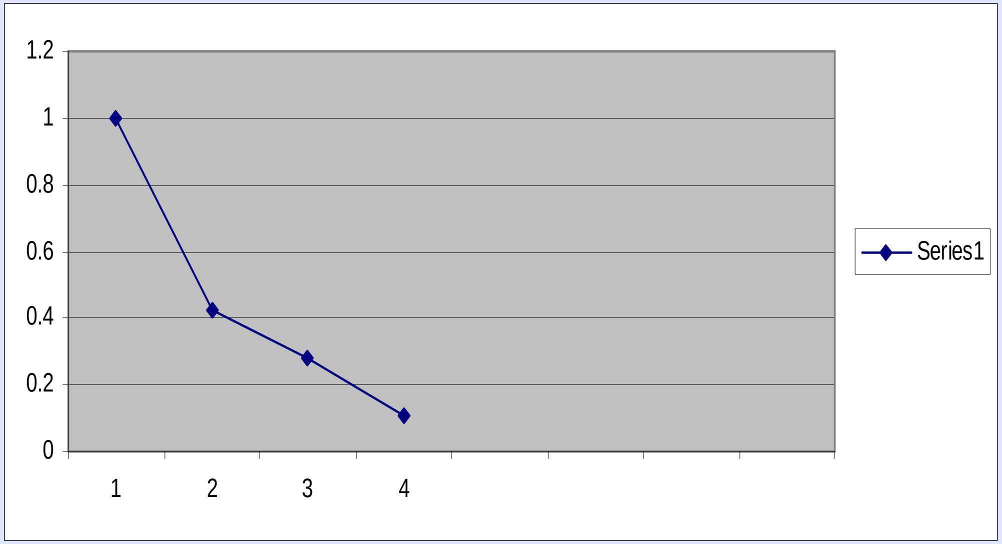
- When the number of processors increases the process time increases, this was not the desired effect.
- This is because of the problem domain. We are sending pixels of the image as the packets of size 76000 to the computers. But the processing of 76000 pixels on one computer takes less time than sending and receiving the data.

Speed up



- The speed up is decreasing as the number of processors is inreasing.
- This shows that the gain we achieve by parallelizing the algorithm is very low, lower than 1.
- So serial processing is faster than paralel processing for this project.

Efficiency



- The efficiency we get by increasing number of CPU is decreasing.
- If we compile the code by only one CPU then we can get full efficiency, but the efficiency decrease while number of CPU increases.
- In other words, efficiency is not proportional to CPUs more than a specified number of CPU.

As a result the speed up and efficiency is decreasing, the process time increases as the number of processers is increased all this charts shows that the edge detection is not feasible for parallel processing.