HPC in Chile academy

Roberto Gonzalez
PhD Astrophysics Chile MetricArts

Computer Clusters for Astronomy in Chile

	TOP 1 Summit IBM (USA) V100 GPU	Leftraru (Uchile) multipurpose	Geryon2 (PUC) dedicated	Kultrun (UdeC) dedicated
Cores	2397824	2640	728	480
Performance (Tflops)	143500	70	22	14
Memory (TB)	2800	6.25	7	3.5
Arch: nodes Proc/cores	4608 Nodes 6 Nvidia Volta V100 + 2 Power9 cpu per node	132 Nodes. 2x10 cores	15 Nodes 4x10 cores 4 Nodes 4x8 cores	8 Nodes 2x16 cores 1 Node 224 cores (shared mem)

Computer Clusters for Astronomy in Chile

Software:

- Linux distributions (Centos/Redhat)
- Torque/Maui Resource manager and Cluster Scheduler
- Compilers: gnu C, C++,fortran + OpenMPI or Intel MPI + threads/OMP
- Scientific libraries cublas, fftw, gsl, etc
- Python + most science packages + mpi4py + threads
- No graphical interface

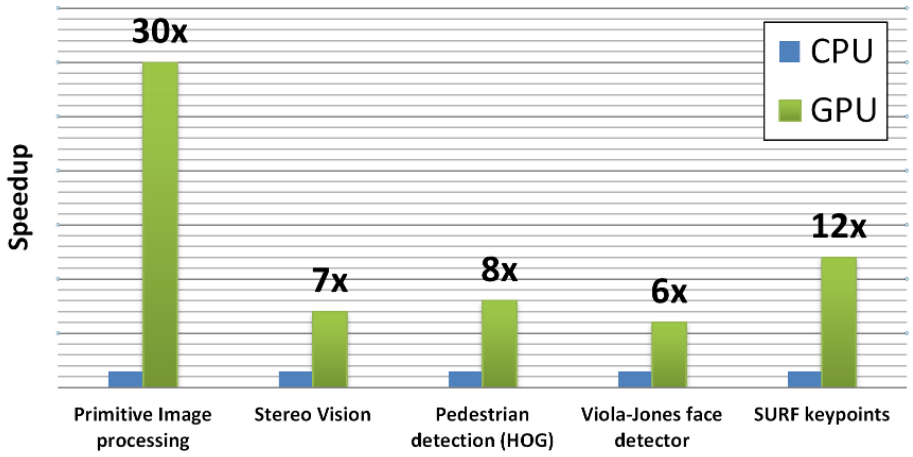
Little Usage:

- Docker containers
- Virtual Machines + other parallel platforms Hadoop, Spark
- Tensorflow

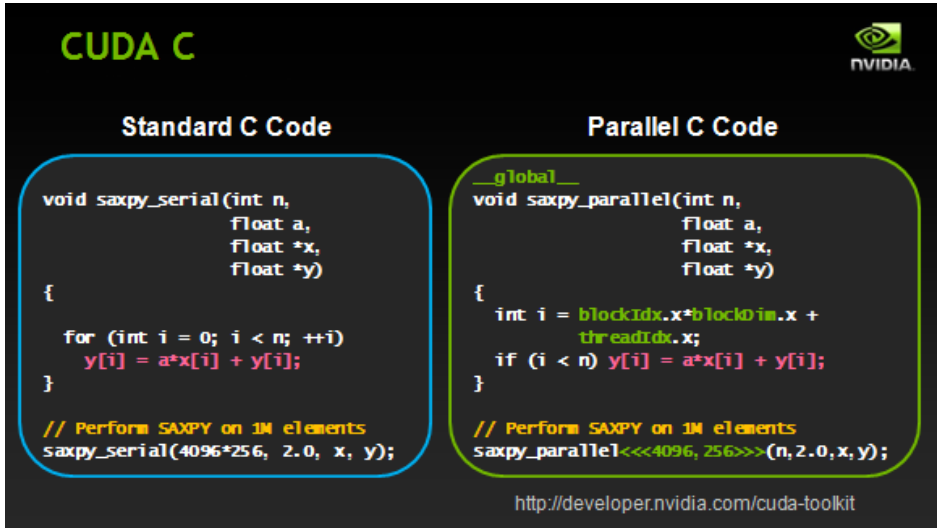
GPU in HPC for image processing

Most image processing in astronomy is made using cpu (IRAF, Sextractor, etc)

CUDA performs 2D Fast Fourier transforms
10x – 100x faster than CPU in a medium tier card



FFT Execution time	CPU	GPU
4K image	~1 s	30ms
Gigapixel image	~200 s	1 s
Mid-tier Nvidia GPU Card vs CPU i7		



Basic Concepts

- **HPC**: High performance computing
- **CPU/Processor/Core**: All three different definition but for practical case it will be the minimal processing unit to run a single serial code or task.
- **Node**: Comprise of multiple CPUs/Processors/Cores, physically in the same 'box', and share same memory and data bus. (i.e. dual- quad- core intel desktop computers).
- **Task**: typically a program or code which can be run in a single processor, a parallel program consist of multiple tasks running on multiple processors.
- **Thread**: similar to a task, but in the context of pieces or entire tasks allocated in an execution queue, and different treads can be run in different processors or even in the same, in serial or parallel, and share the same data/memory.
- **Communication**: Where different task send/receive data through the bus (in the case of processors in a node), or through network (between nodes, drawers, etc)
- **Synchronization**: Is the coordination between different tasks, usually refers to the case when one task must wait other task to finish and communicate to resume execution.
- **Granularity**: Is a qualitative concept of the ratio of computation and communication/synchronization. A Fine granularity means, means most work and time spent is used in computation, while a Coarse granularity means a large amount of work and time is used for communications/synchronization.
- **Observed Speedup**: One of the simplest indications for parallel performance, it is just the ratio between the serial execution time of a program and the parallel execution time.
- **Vectorization**: Automatic parallelization of some loops and array processes made by the compiler.

Basic Concepts

- **Parallel overhead**: Amount of code and time required to prepare and coordinate parallel task, it is not useful work in short words.
- **Scalability**: Refers to a parallel program's ability to demonstrate a proportional increase in parallel speedup, with the addition of more tasks. Usually, overhead, granularity, synchronization and communication are the main factors which contribute to scalability.
- **Potential speedup or Amdahl's Law**:

Is the ideal speedup a program can reach if we know the fraction of the program P that can be parallelized

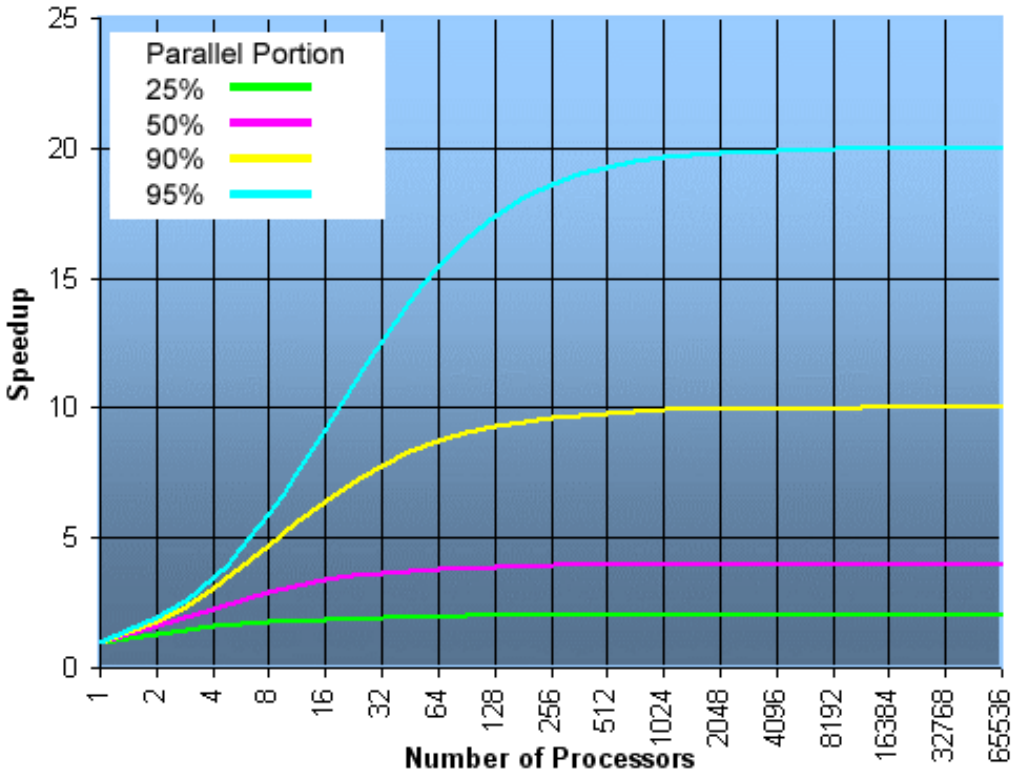
Max Speedup = $\frac{1}{1-P}$

For P-1 the speedup is infinite(ideal case)

For a number of processors/tasks N, and Serial fraction S we have

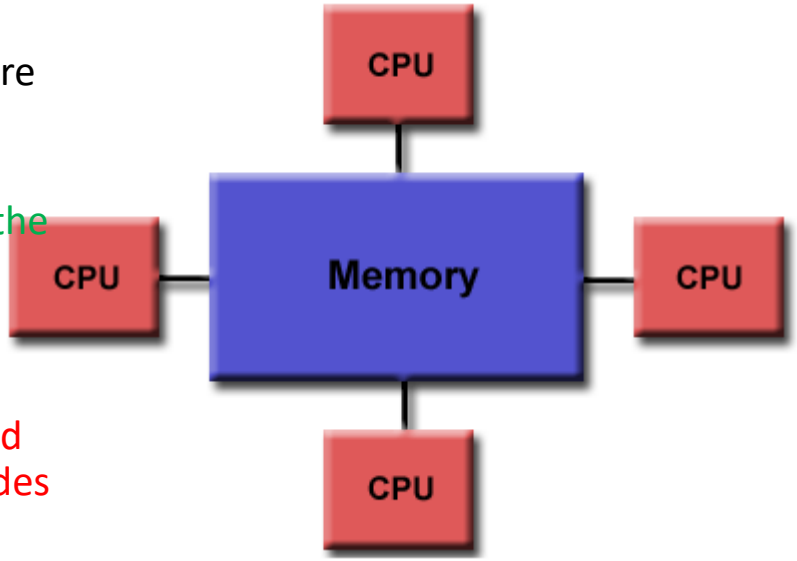
Speedup = $\frac{1}{\frac{P}{N}+S}$

Challenge for massive paralleling!!!



Basic Concepts

- **Shared Memory**: Is a system with the architecture where all processors have direct access to a common physical memory.



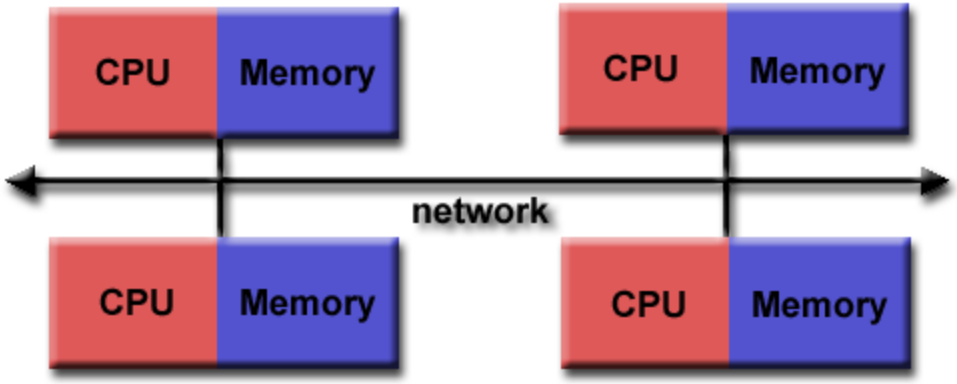
-Changes in memory made by one processor are visible by the others. Easy communication and programming.

-Fast data sharing due proximity of memory to CPUs

-Lack of scalability between memory and CPUs

-It become increasingly difficult and expensive design shared memory machines with larger number of CPUs. (Typical nodes reach 8-16 cpus.)

- **Distributed memory**: Memory is local for each cpu, and require a network for communication.



-Memory scale with number of cpus.

-Each processor have exclusive en fast access to own memory

-Cheap

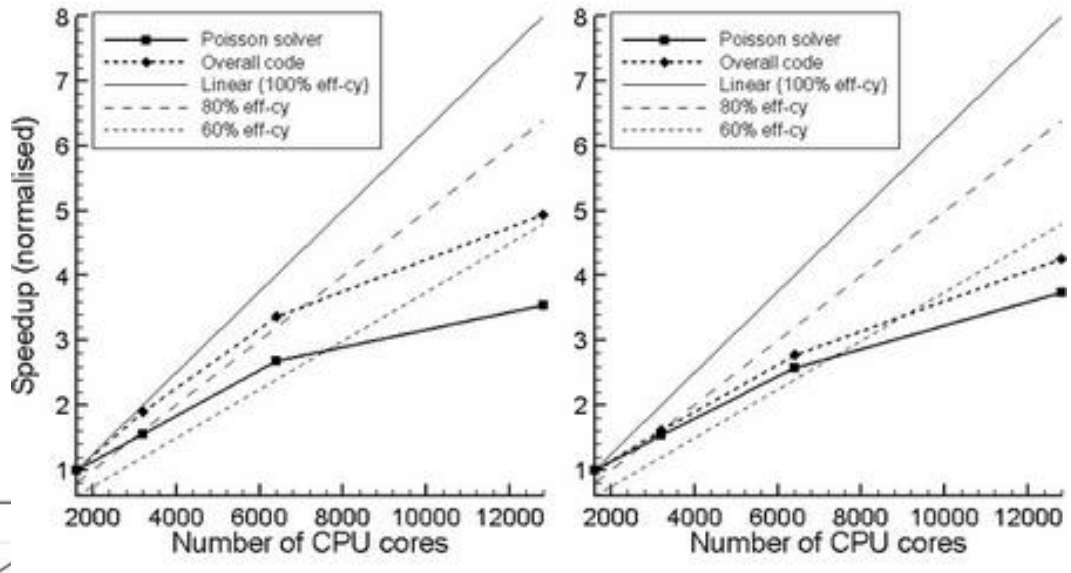
-Communication scheme is responsibility of the programmer

-Network speed usually slower than a local data bus.

Typical computer clusters are a combination of these architectures.

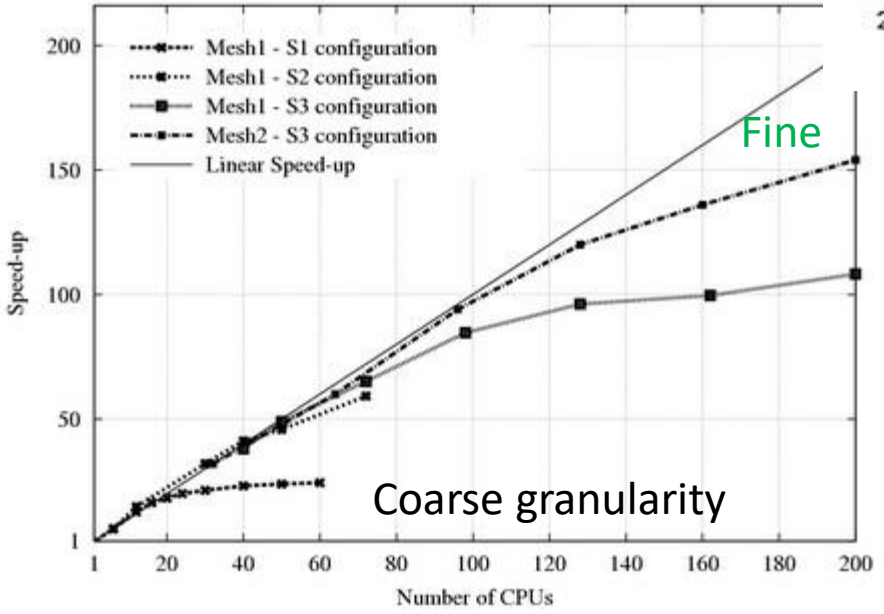
Scalability: Speedup

Parallel Poisson solver using PM code

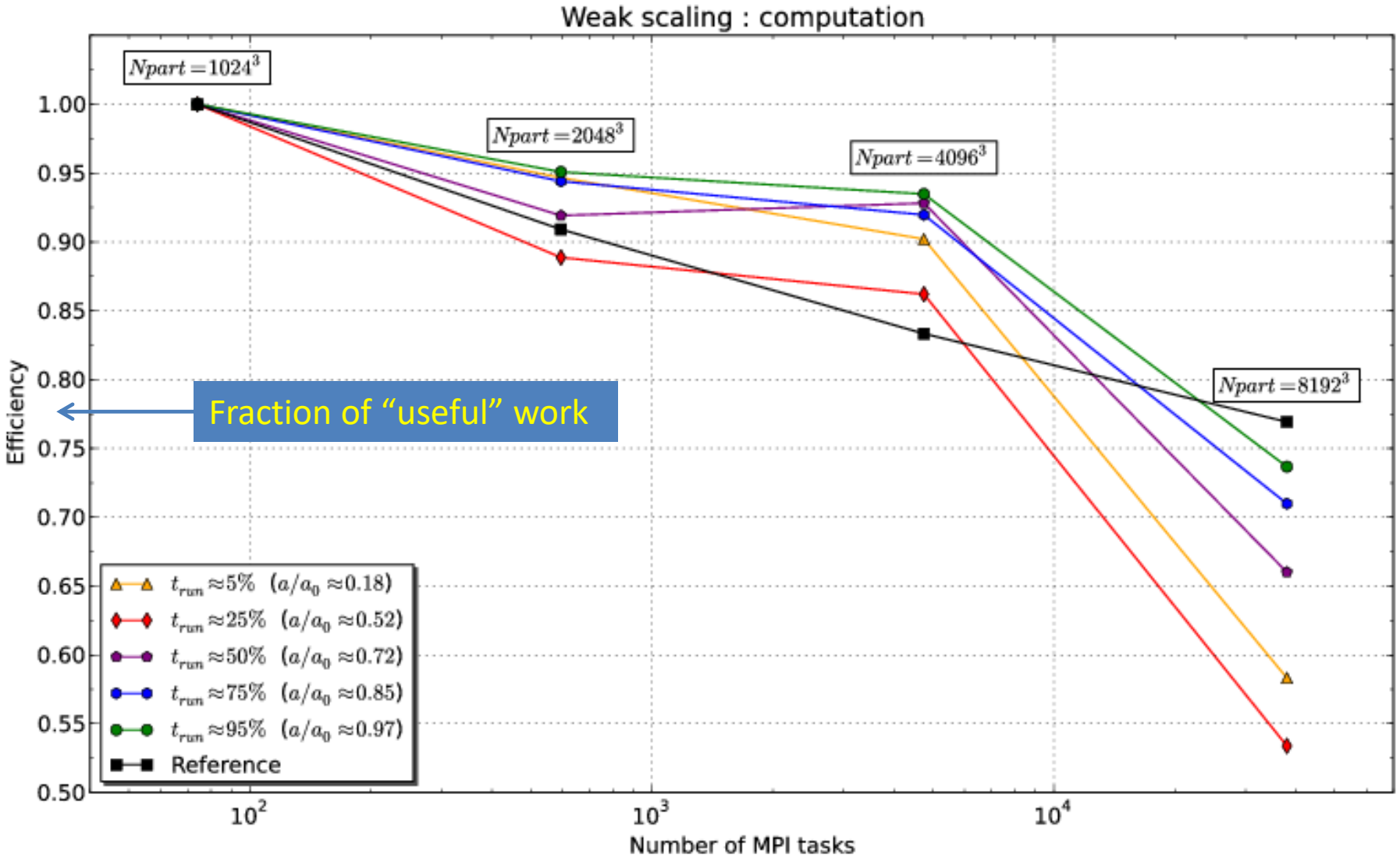


Different parts of the code scale at different ratios.

Speedup different particle distributions



Scalability: Efficiency

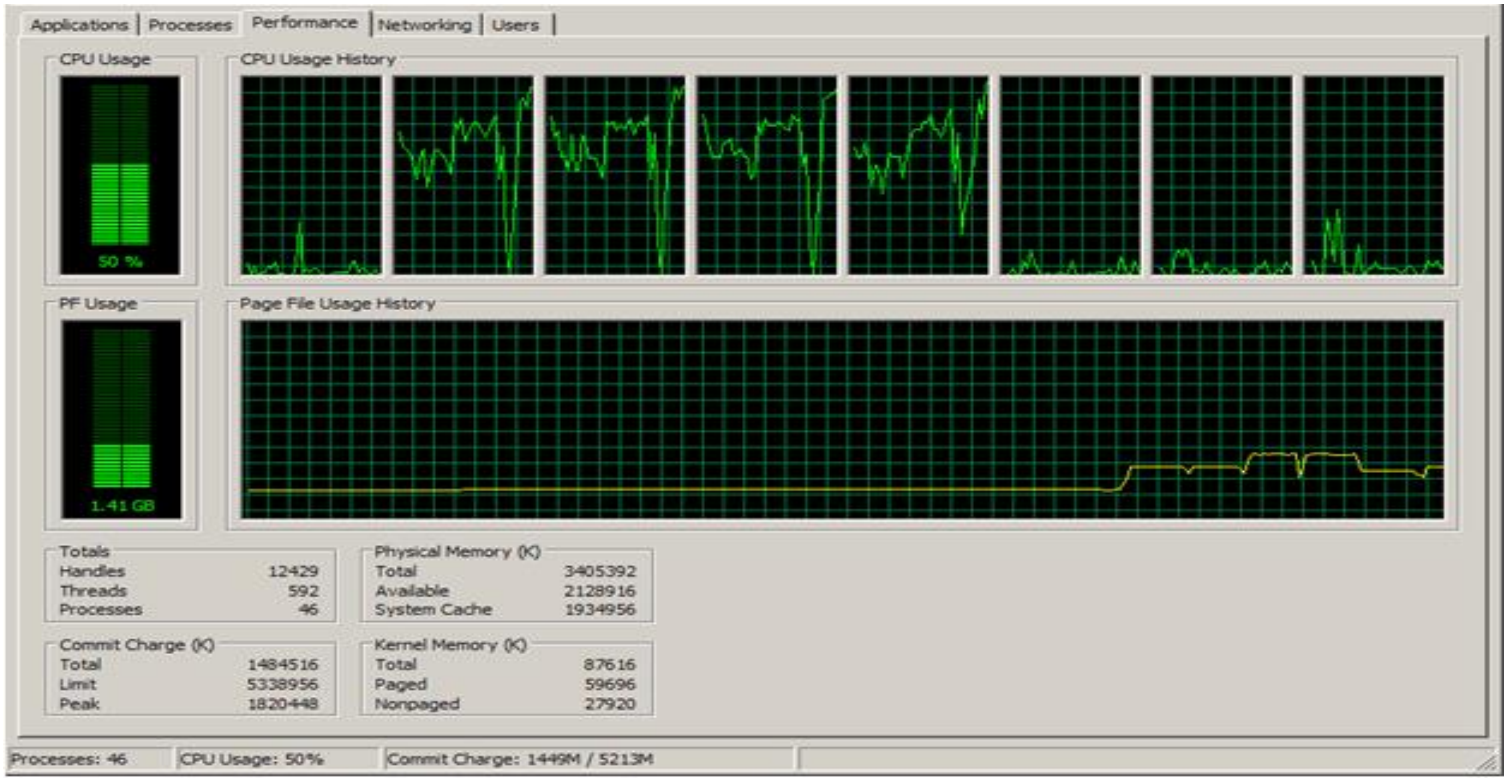


Load balance

Is the measure of the differences in the distribution of work among all the tasks of our parallel program for a given time range.

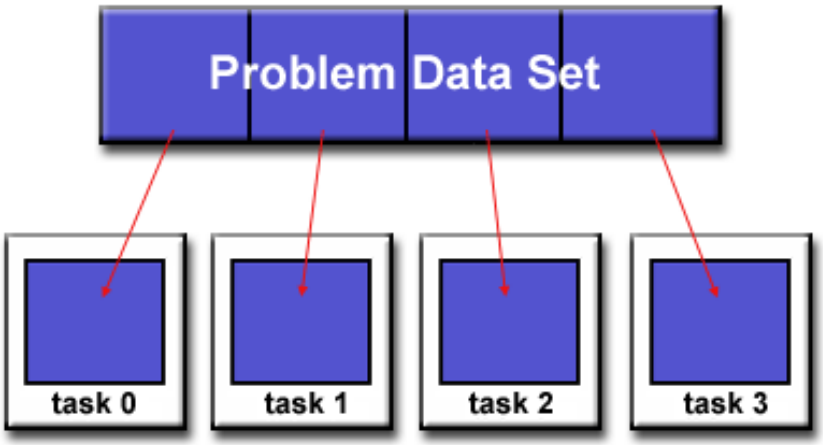
Perfect/Optimal load balance : Is the case when the total computation work is equally split among all tasks

Poor load balance : When the work of one or a few task is much larger than the average. This deprecates directly in the scalability, and the total execution time will depend on the execution time of the most loaded processors rather than scale with the number of task

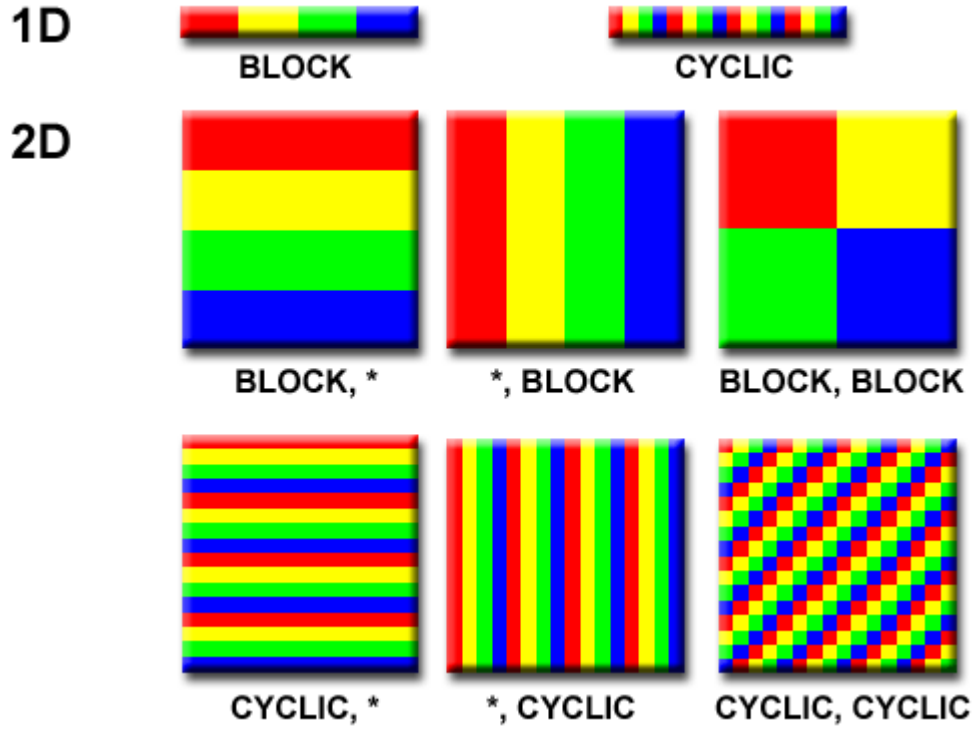


Domain Decomposition

- Goal : assign work to each task to achieve perfect load balance



Typical for arrays



Domain Decomposition