ISUM 2015 6th INTERNATIONAL SUPERCOMPUTING CONFERENCE IN MEXICO
Mexico City, Mexico
March 9-13, 2015
www.isum.mx

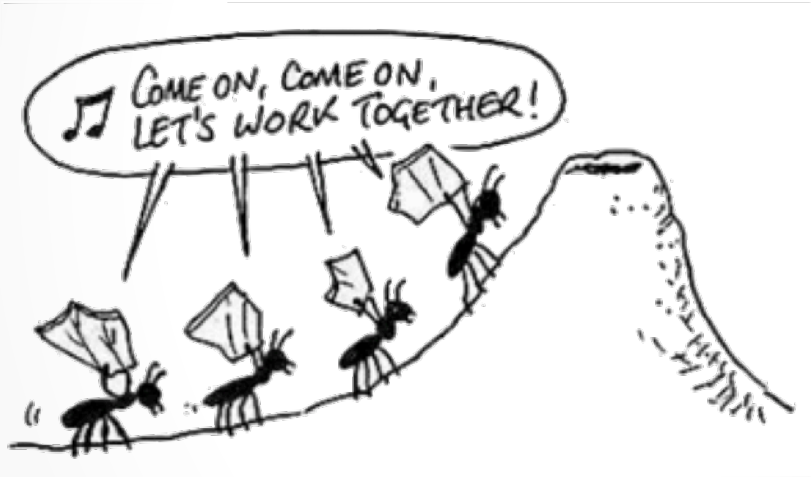
An agglomeration strategy for the parallel process mapping on a distributed computing architecture

M.C. Juan Carlos Catana-Salazar
Dr. Jorge Luis Ortega-Arjona

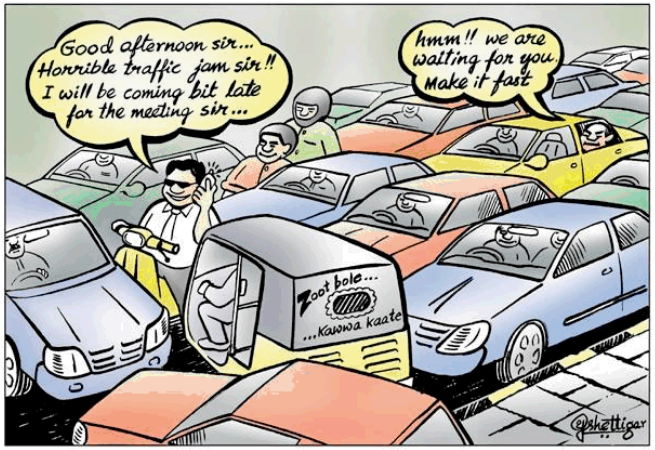
Posgrado en Ciencia e Ingeniería de la Computación
Universidad Nacional Autónoma de México

1. Intro

- *Parallelism*



Parallel System



Communication & Sync

1. Intro

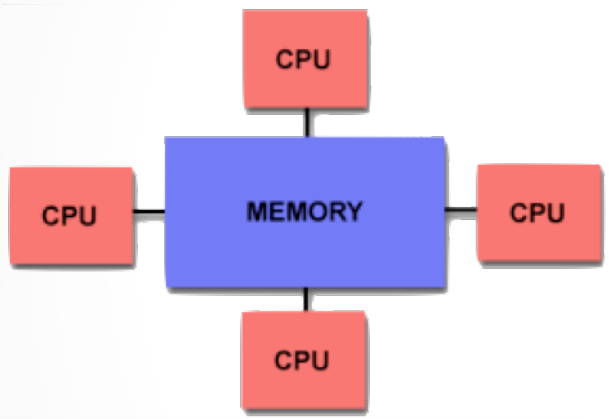
- *Granularity*

Coarse grained

Fine grained

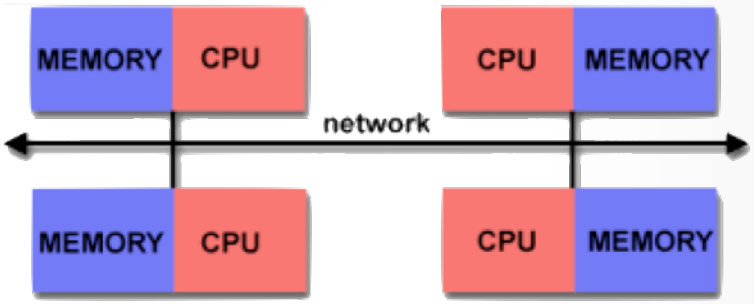
1. Intro

- *Types of communications*



Shared memory

$t_c \approx$ constant



Distributed memory

t_c = variable

1. Intro

$$t_{constant} \ll t_{variable}$$

$$ET(P) = PT(P) + \underbrace{c_l(P) * t_{constant} + c_r(P) * t_{variable}}_{\textbf{Communication time}}$$

2. The Problem

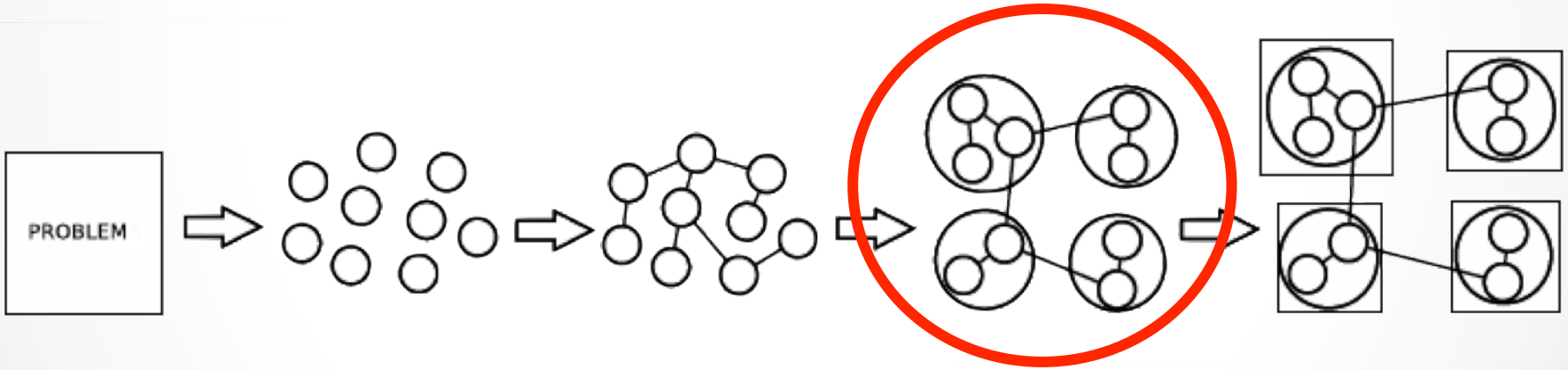
It is possible to minimize communication time?

Communications are needed for parallelism, how can it be minimized?

3. Background

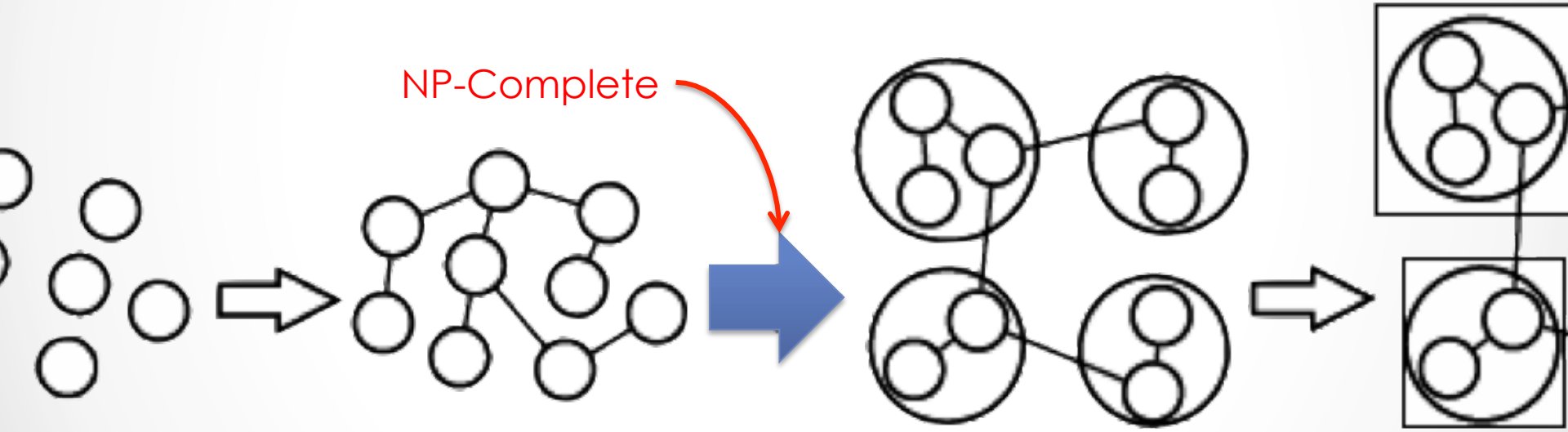
- Parallel software design

Agglomeration



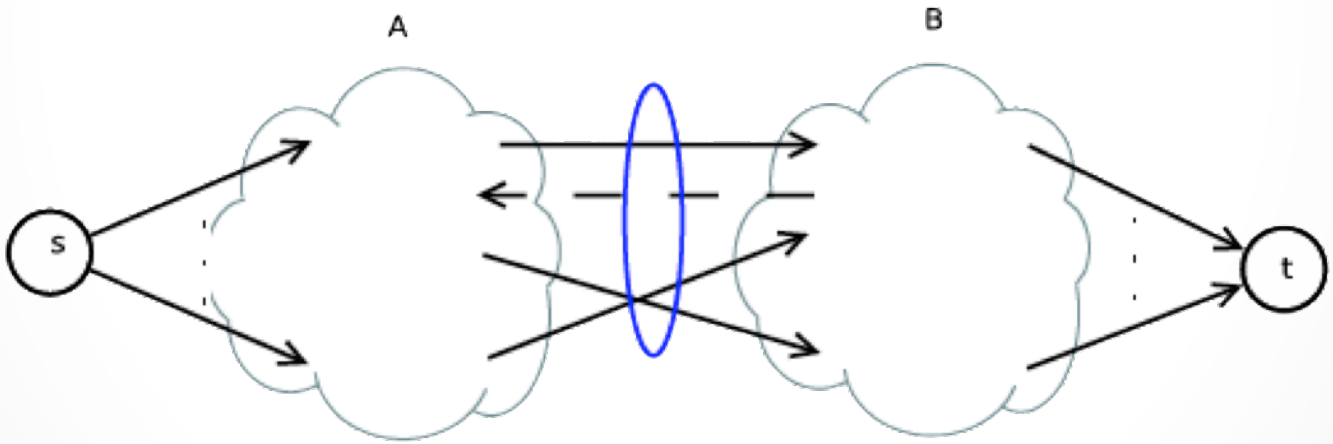
3. Background

- How to get partitions from the software graph?



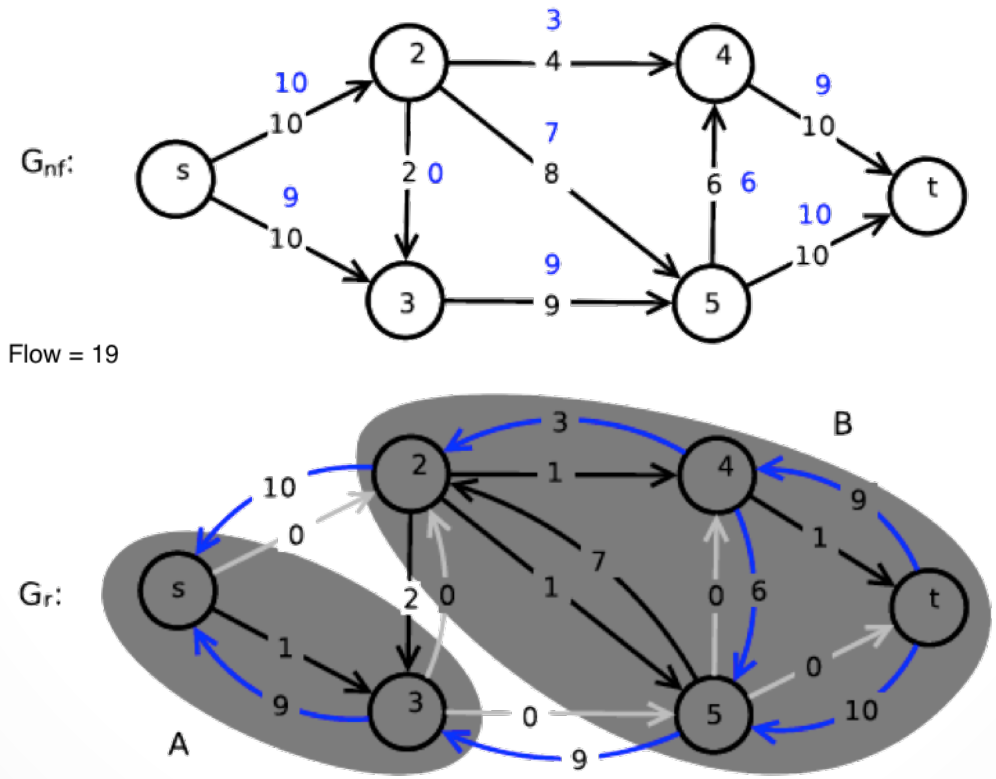
3. Background

- Happily, there is something called "**Max Flow-Min Cut Theorem**"

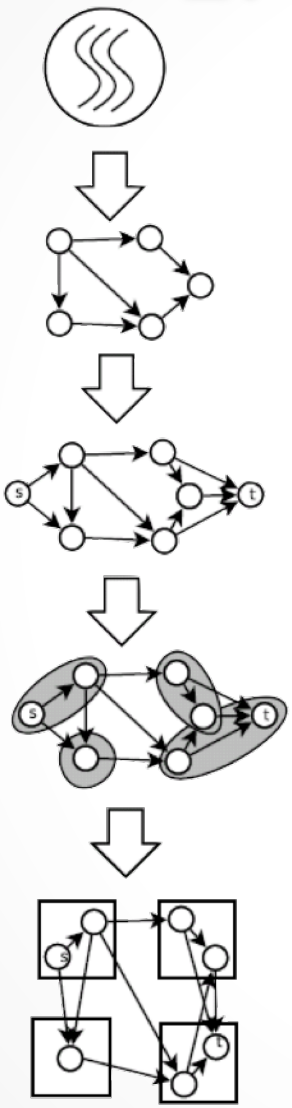


3. Background

- How does it look?



4. The Strategy



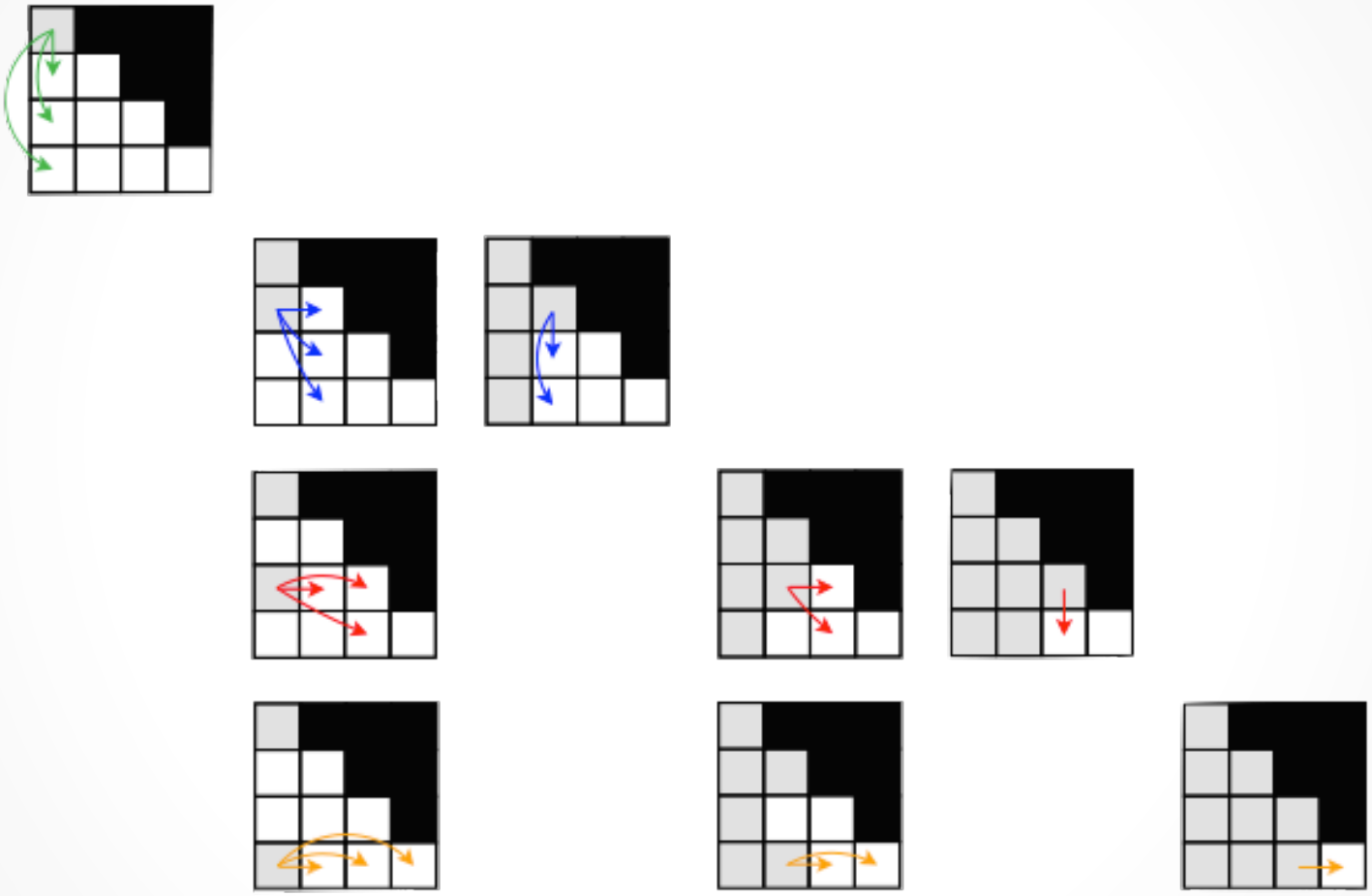
Building the Software Graph

Transformation to a Network Flow

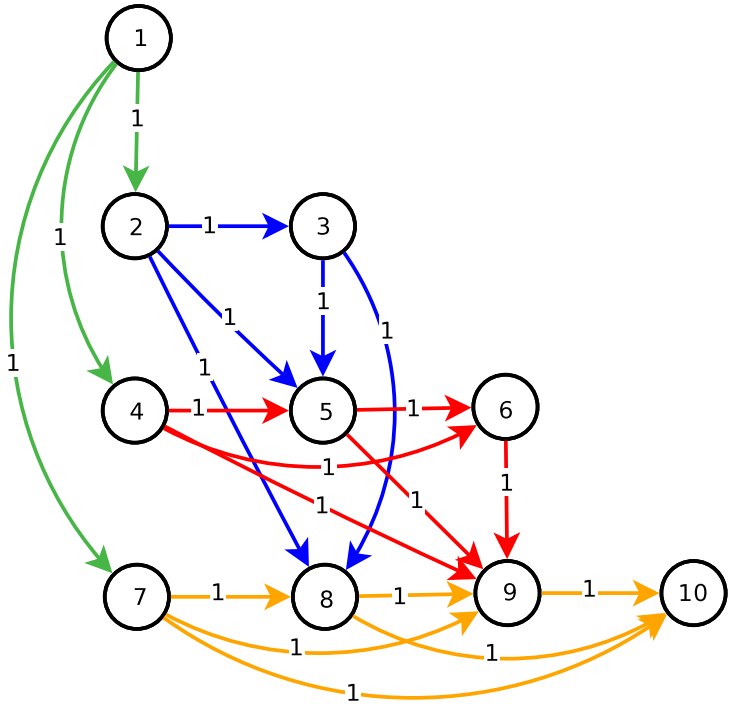
Applying the Algorithm

Mapping in the nodes

4.1 Building the graph

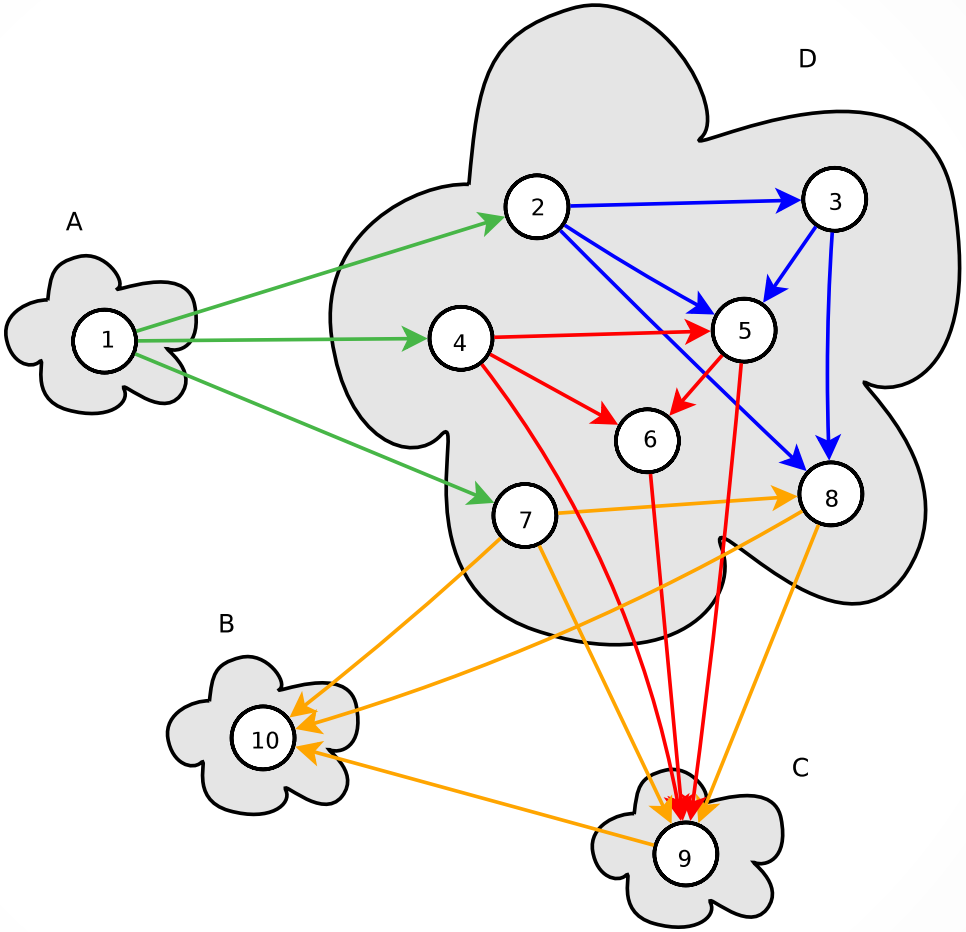


4.2 Transform to a Net Flow



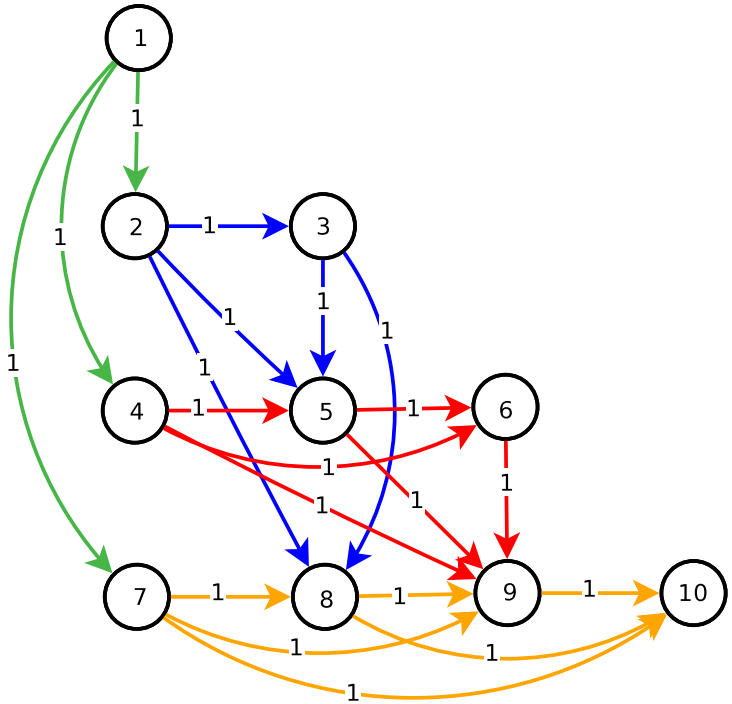
Comms = 20 units

4.3 Applying the algorithm



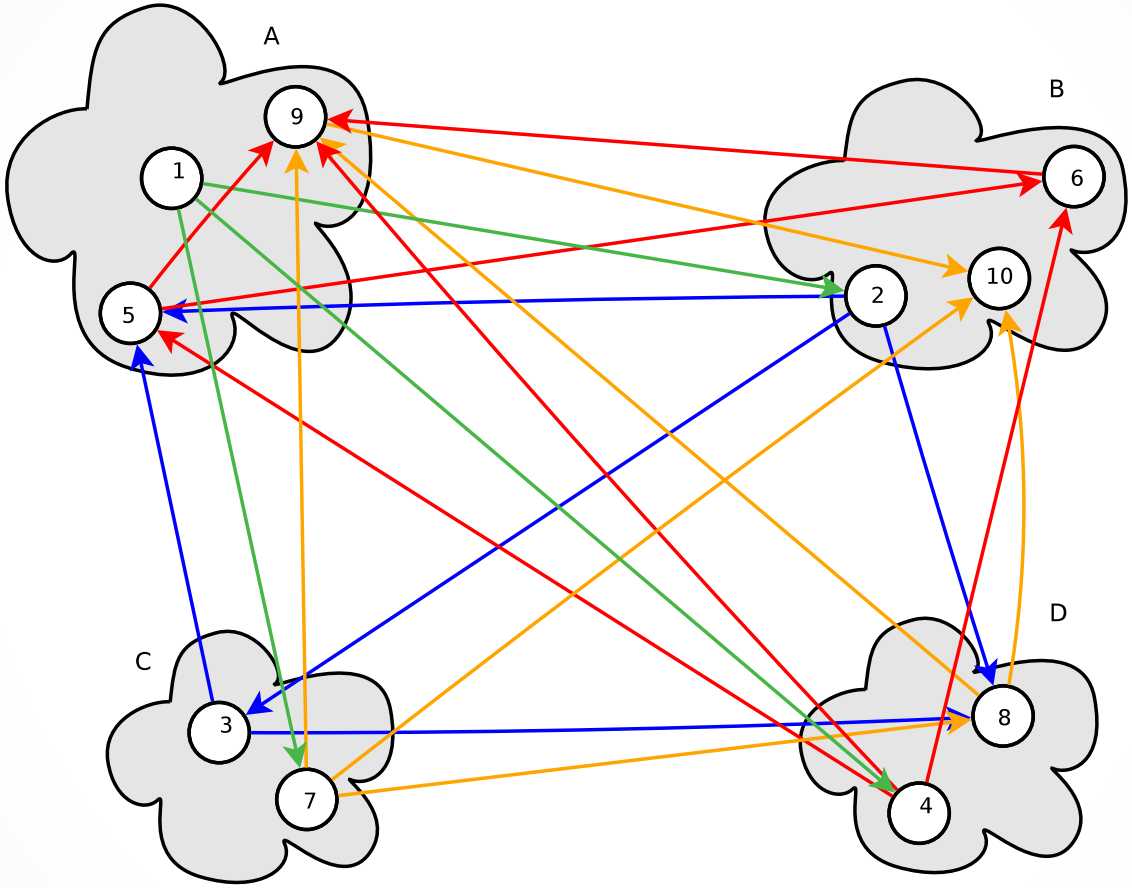
$$CT = 9 * t_{constant} + 11 * t_{variable}$$

5. Comparison



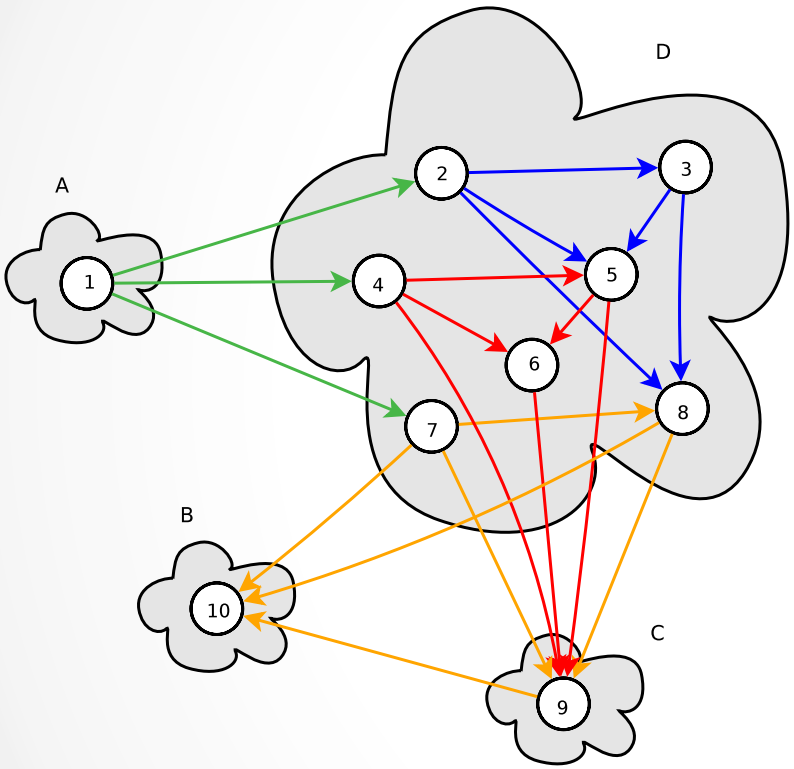
Comms = 20 units

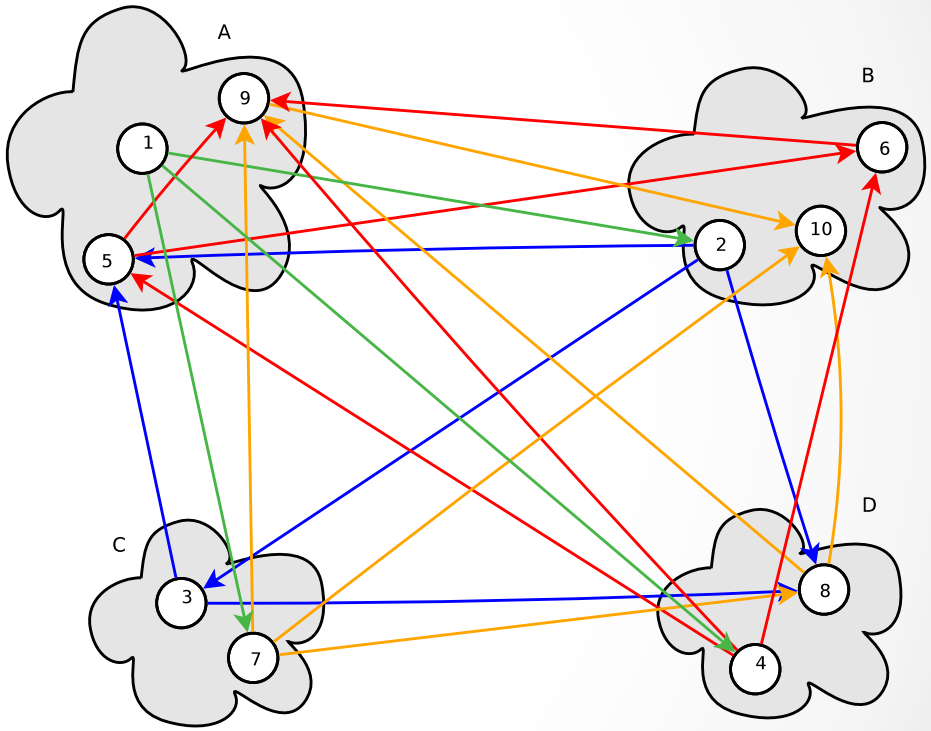
5. Comparison



$$CT = 1 * t_{constant} + 19 * t_{variable}$$

5. Comparison





$CT = 9 * t_{constant} + 11 * t_{variable}$

$CT = 1 * t_{constant} + 19 * t_{variable}$

5.1 Experimental data

	Data set	RR	NF	Porcentaje
1	4 x 4	2,226.8	2,380.7	-6.9%
2	16 x 16	2,271.9	2,387.2	-5.08%
3	64 x 64	2,258.2	2,399.2	-6.24%
4	512 x 512	2,592.8	2,677.5	-3.2%
5	2048 x 2048	5,038.7	5,492.7	-9%
6	8192 x 8192	104,705	103,833.4	0.83%
7	16,384 x 16,384	779,318	761,796.4	2.25%

4 nodes each with 8 cores
16 GB RAM
OS Ubuntu 12.04 LTS (GNU/Linux 3.2.0-27-generic x86 64)
Network switch Gigabit Ethernet

6. Conclusions

- A parallel system's performance is directly affected by the amount and type of communication of its processes.

- Minimize remote communications and maximize local communications benefits the performance of a parallel system.

6.1 Not fairly clear

- How much benefit gives a mapping based in communications than a mapping based in load balancing?
- If we could guarantee load balancing and minimum amount of remote communications in k-partitions, does that would result in an optimal execution time?

7. Future work

- Detailed and extended experimental work.
- Provide an algorithm for *k*-minimum partitioning.
- Extend or provide an algorithm considering both properties (load balancing and remote comms).
- Clarify the statements in the previous slide.

9. Refs

- K. Mani Chandy, Stephen Taylor, *An Introduction to Parallel Programming*.
- Almasi, G.S. and A. Gottlieb, *Highly Parallel Computing*.
- Ian Foster, *Design and Building Parallel Programs*
- Blaise Barney, *Introduction to Parallel Computing*.
- Jon Kleinberg, Eva Tardos, *Algorithm Design*.
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest,Ronald L., *IntroductiontoAlgorithms*.