Slide 1

Load Balancing and Priority of Tasks in Distributed Environments

Salah DOWAJI and Catherine ROUCAIROL
"Parallélisme Non Numérique" (PNN) Team.
University of Versailles-St Quentin
PRiSM laboratory
45, avenue des Etats-Unis
78035 Versailles Cedex - FRANCE
e-mail: {sad,roucairo}@prism.uvsq.fr

Slide 2

Outline

1- Issues related to the Branch-and-Bound (B&B) algorithm.

2- Application: Vertex Cover Problem (VCP).

3- B&B approach for the VCP under PVM.

4- Developed load balancing (LB) strategies.

5- A new notion of priority between subproblems

- Motivations & justifications
- Definition of potentiality notion

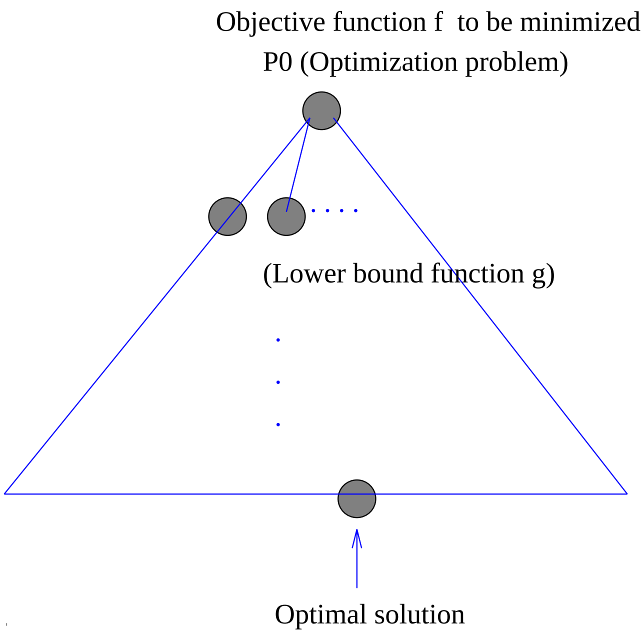
6- Influence of using the potentiality notion on:

- Load distribution,
- Communication costs.

7- Conclusion & perspectives.

Slide 3

B&B algorithm (nature)



Slide 4

B&B algorithm (underlying idea)

The **lower bound function *g*** for a subproblem gives the smallest possible cost of a solution to that subproblem. Its values satisfy the following conditions:

- $g(P_i) <= f(P_i)$ for P_i a subproblem in the tree,
- $g(P_i) = f(P_i)$ for P_i a terminal subproblem,
- $g(P_i) >= g(P_j)$ if P_j is a descendant of P_i.

The value $g(P_i)$ is generally called the **partial cost** of P_i.

Slide 5

B&B algorithm (procedures)

A typical B&B algorithm consists of four major procedures:

Selection chooses a single subproblem from the generated subproblems not yet examined. The following heuristic search strategies are commonly used:

- Best-first,
- Depth-first, or
- Breadth-first.

Branching generates from a subproblem new smaller subproblems, and calculates their lower bounds.

Elimination prunes a newly created subproblem if its lower bound (partial cost) is greater than or equal to that of the incumbent (value of current solution).

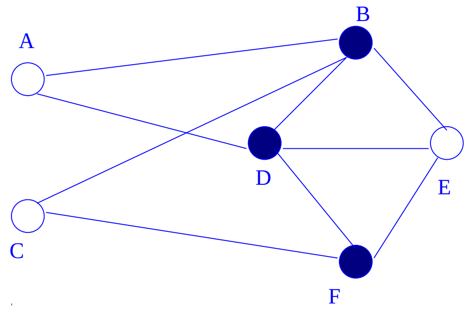
Termination Test checks whether there is no more subproblems to be examined.

Slide 6

Vertex Cover Problem (VCP)

Definition
For an undirected graph $G = (V, E)$
$U \subseteq V$ is called a vertex cover if every edge $u, v \in E$ is covered by a node in it U ($u \in U$ or $v \in U$)
VCP aims to find the vertex cover with minimal cardinality (NP-complete).

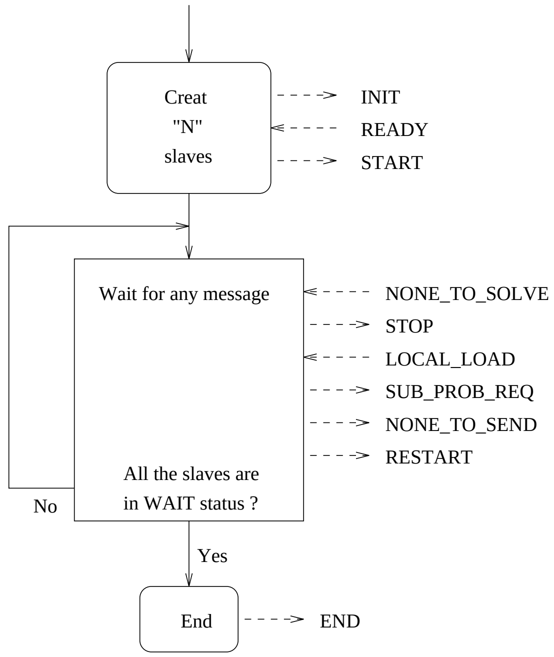
Example:



Slide 7

PVM implementation of B&B VCP (1)

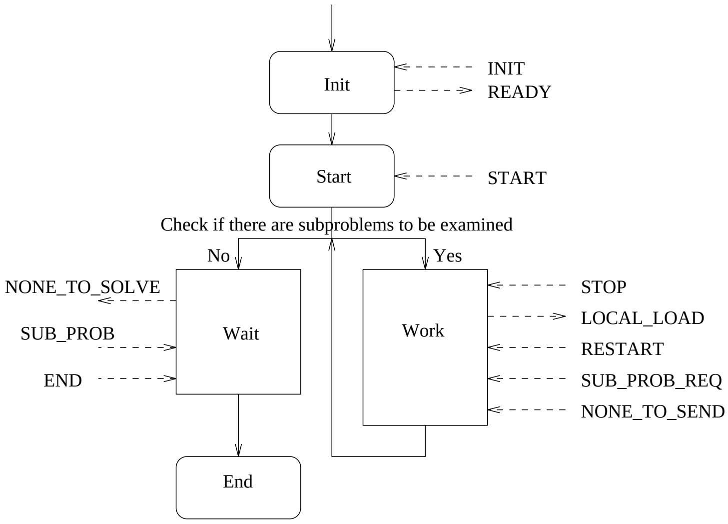
Master algorithm



Slide 8

PVM implementation of B&B VCP (2)

Slave algorithm



Slide 9

Developed LB strategies

We developed VCP with three load balancing strategies:

One-by-one Each idle slave receives only one subproblem from the beginning of the queue of another loaded slave.

Partial distribution (V1) Each idle slave receives several subproblems from **the beginning** of the queue of another loaded slave.

Partial distribution (V2) Each idle slave receives several subproblems from **the end** of the queue of another loaded slave.

Slide 10

Partial distribution LB strategy (example)

	Slave_0	Slave_1	Slave_2	Slave_3	Slave_4
Local load	10	0	8	17	19

Average load = 54/5 # 10 tasks per slave.

Deviation	0	-10	-2	5	9

min. deviation max. deviation

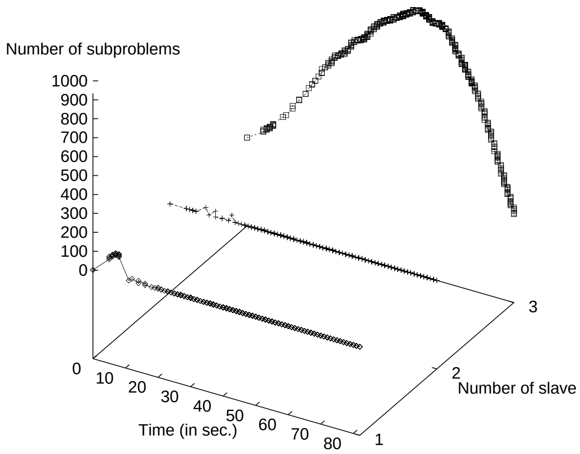
Send min(Average load, (local load - average load))

An example of application of partial distribution load balancing strategy, where slave_1 asks the other slaves for work.

Slide 11

Load distribution (1)

Priority is given by partial cost
Strategy: **one-by-one**
Root problem: Graph of 75 nodes with average degree of 50%



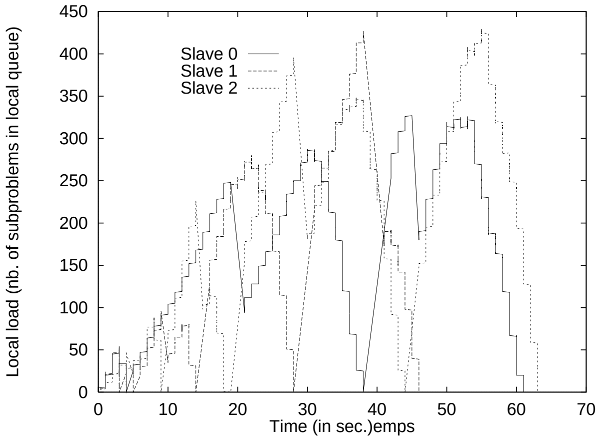
Slide 12

Load distribution (2)

Priority is given by partial cost

Strategy: **partial distribution (V1)**

Root problem: Graph of 75 nodes with average degree of 50%



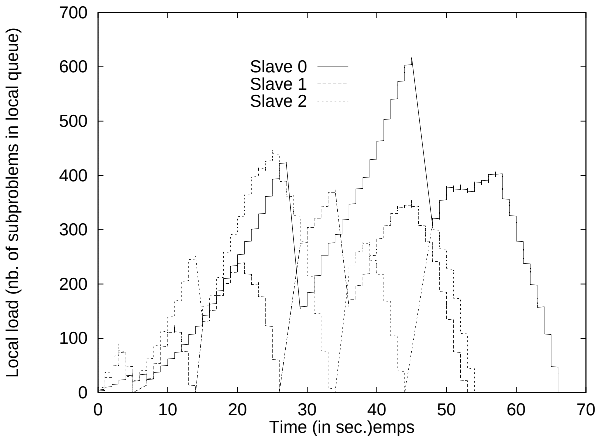
Slide 13

Load distribution (3)

Priority is given by partial cost

Strategy: **partial distribution (V2)**

Root problem: Graph of 75 nodes with average degree of 50%



Slide 14

Discussion & Results

Partial cost is **NOT** the only parameter to define priority between subproblems.
A new notion of priority based on the potentiality of a subproblem sp is defined as the following:

Definition .1

$$Potentiality(sp) = (dist(sp, End), potential\_work, ancestors\_nb).$$

Where:

sp is the considered subproblem,

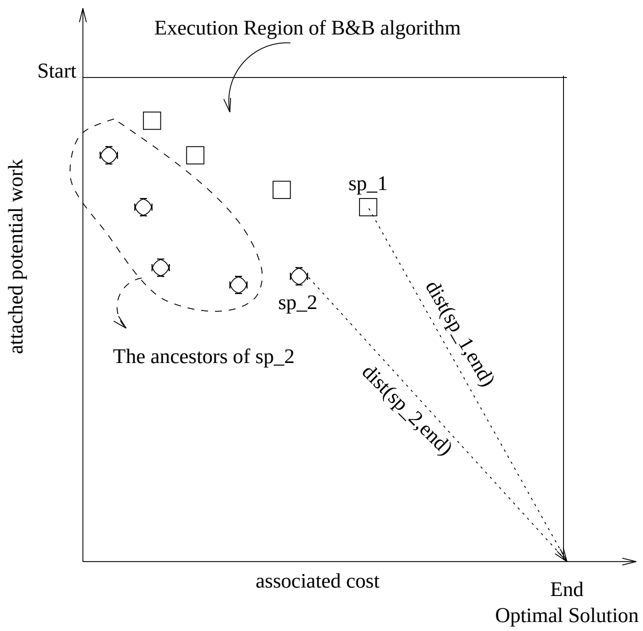
$dist(sp, End)$, the distance between the position of sp and the **End** point,

$potential\_work$, the value of the associated potential work of sp (application dependent),

$ancestors\_nb$, the number of the ancestors of sp.

Slide 15

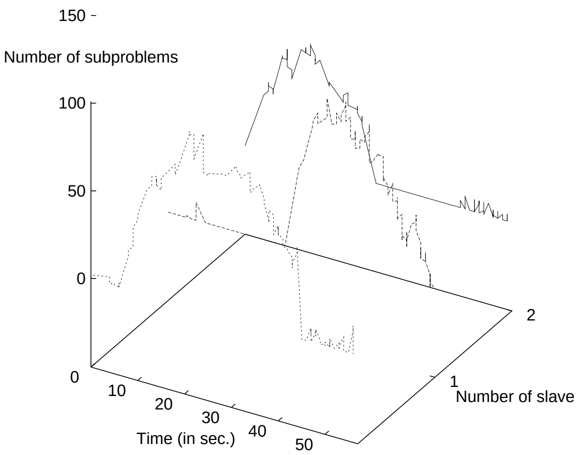
Definition of priority between sp_1 and sp_2



Slide 16

Influence on load distribution (1)

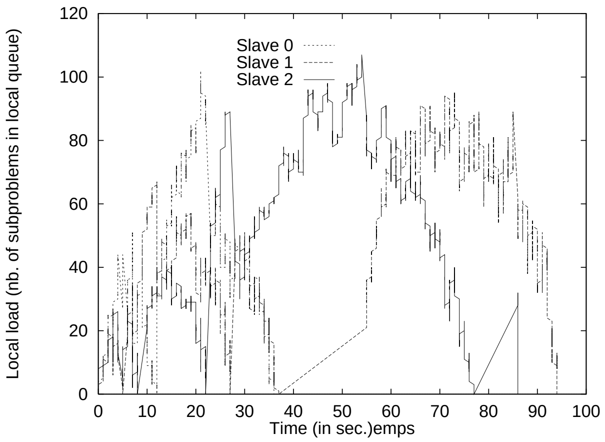
Priority is given by the potentiality notion
Strategy: **one-by-one**
Root problem: Graph of 75 nodes with average degree of 50%



Slide 17

Influence on load distribution (2)

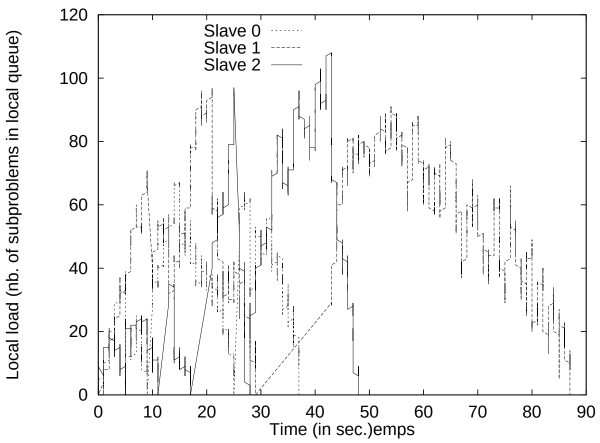
Priority is given by the potentiality notion
Strategy: **partial distribution (V1)**
Root problem: Graph of 75 nodes with average degree of 50%



Slide 18

Influence on load distribution (3)

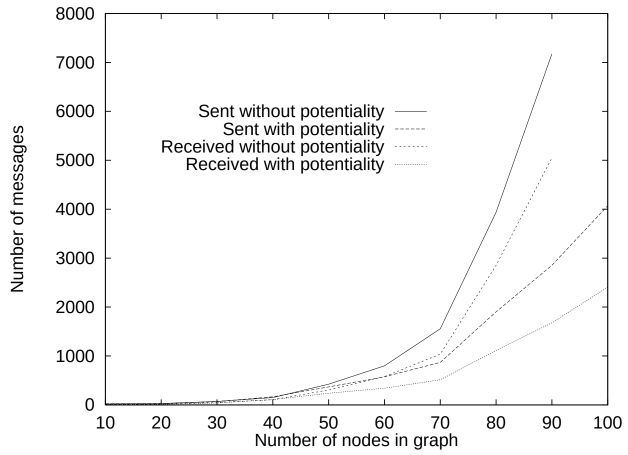
Priority is given by the potentiality notion
Strategy: **partial distribution (V2)**
Root problem: Graph of 75 nodes with average degree of 50%



Slide 19

Influence on communication costs (1)

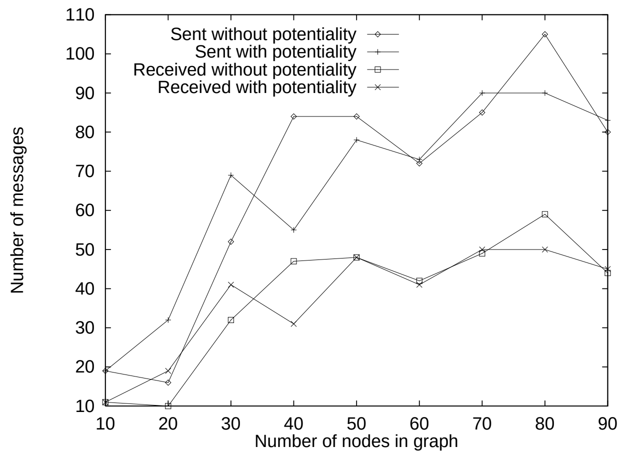
Strategy: **one-by-one** strategy



Slide 20

Influence on communication costs (2)

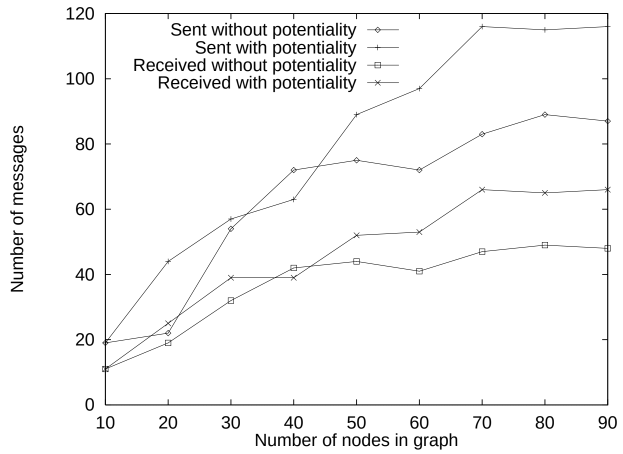
Strategy: **partial distribution (V1)** strategy



Slide 21

Influence on communication costs (3)

Strategy: **partial distribution (V2)** strategy



Slide 22

Conclusion & perspectives

We analyzed the behavior of three load balancing strategies.
We showed the necessity to have a new priority order between subproblems.
We introduced a new notion of priority between subproblems (notion of potentiality).

Future works will consist of:

- investigating the use of our potentiality notion when we have several masters instead of only one.
- testing the validity of our potentiality notion with other B&B applications.