A First Step to the Evaluation of SimGrid in the Context of a Real Application

Abdou Guermouche and Hélène Renard,

LaBRI/Univ Bordeaux 1 – I3S/École polytechnique universitaire de Nice-Sophia Antipolis

April 19, 2010

Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

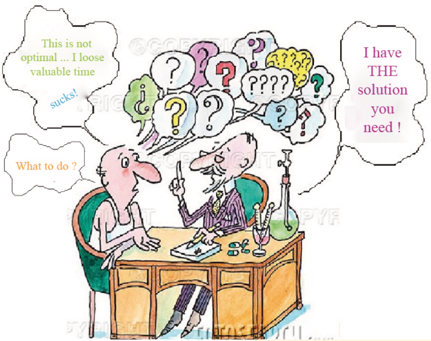
Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

Data redistribution algorithms : context


This is not optimal ... I loose valuable time
sucks!
What to do ?
I have THE solution you need !

Target platforms: distributed heterogeneous platforms (network of workstations, clusters of clusters, grids, etc.)

1. Various sources of load imbalance : application requirements / platform.
2. The data must be redistributed to achieve a better load balancing.
3. No discussion of the mechanism of load balancing we consider it as given.

Data redistribution algorithms : context

- The algorithm operates on a wide array of rectangular sample data:
  - ➥ The array is split in vertical slices;
- This geometric constraint recommends that processors must be organized as a virtual ring:
  - ➥ Each processor only communicates twice (once with each neighbor).

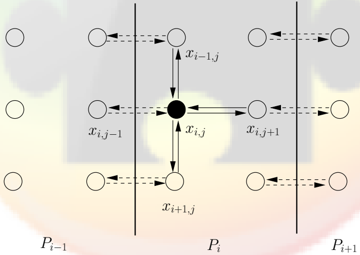


Figure: Communication scheme.

Redistribution problem for heterogeneous bidirectional rings

Definition

A redistribution is light if each processor initially owns all data that it will send during the execution of the algorithm.

$$\text{Minimize } \tau \text{ subject to}$$
$$\begin{cases} \mathcal{S}_{i,i+1} \geq 0 & 1 \leq i \leq n \\ \mathcal{S}_{i,i-1} \geq 0 & 1 \leq i \leq n \\ \mathcal{S}_{i,i+1} + \mathcal{S}_{i,i-1} - \mathcal{S}_{i+1,i} - \mathcal{S}_{i-1,i} = \delta_i & 1 \leq i \leq n \\ \mathcal{S}_{i,i+1} c_{i,i+1} + \mathcal{S}_{i,i-1} c_{i,i-1} \leq \tau & 1 \leq i \leq n \\ \mathcal{S}_{i+1,i} c_{i+1,i} + \mathcal{S}_{i-1,i} c_{i-1,i} \leq \tau & 1 \leq i \leq n \end{cases} \tag{1}$$

To lead to ...

We can use the solution of System 1 safely.

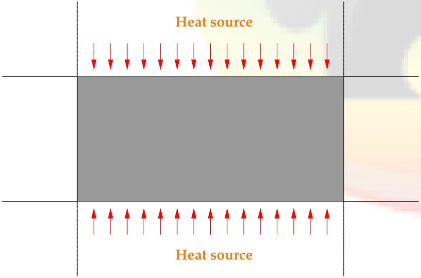
Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

Laplace equation

Context

- *A metal plate to which is applied a source of heat from the edges.*
- *The heat will spread within plate.*
- *The temperature at the edges is kept constant, the heat distribution in the plate tends to a stationary state.*

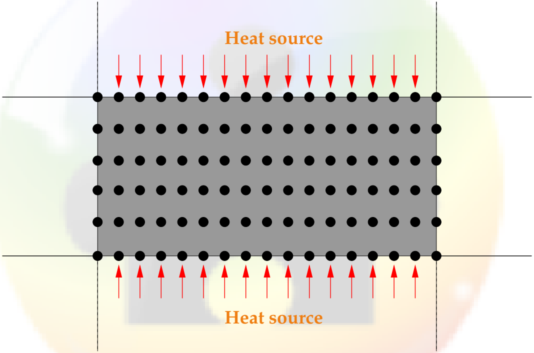


Laplace equation : $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$

Laplace equation : $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$

Resolution :

1. Approximating the solution ⇒ discretization ⇒ grid n^2 points



2. Using finite differences on the Laplace equation, this is equivalent to iteratively solve the following equation:

$$4x_{i,j} - (x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}) = 0$$

Laplace equation: $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$

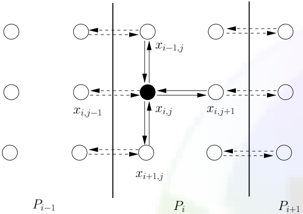


Figure: Communication scheme.

- ➥ Same pattern of communication as the ring of processors
- ➥ Communication only with immediate neighbors.

3. Solving a linear system
   - ➥ Jacobi, since it is of the form: $Ax = b$, with A and x as

$$\begin{pmatrix} \ddots & & & & \\ & \ddots & & & \\ -1 & -1 & 4 & -1 & -1 \\ & & & \ddots & \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{n,n-1} \\ x_{n,n} \end{pmatrix} = b$$

Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

Grid’5000 vs SimGrid

Goal : Compare the behavior of algorithms for load balancing and data redistribution on two different ≪ platforms ≫:

- Grid’5000
- SimGrid

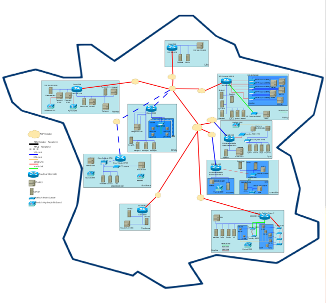


Figure: Grid 5000

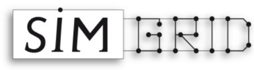


Figure: SimGrid

The *master* and the *workers*

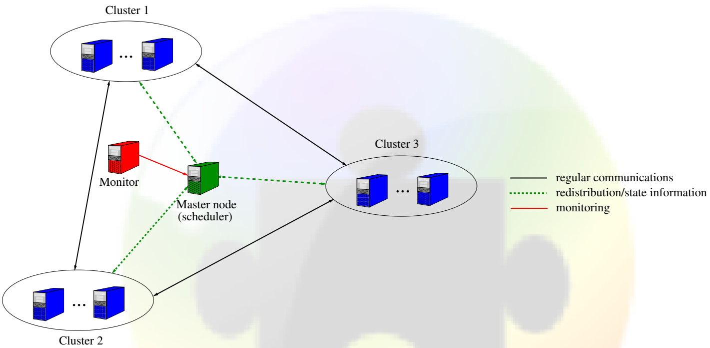


Figure: Experimental scheme: the *master* and the *workers*.

- ➥ This organization is used in both the *simulated* and *real-life* context.
- ➥ The difference comes from the monitor which is given by `SimGrid` in the simulated context.

The *master* and the *workers*

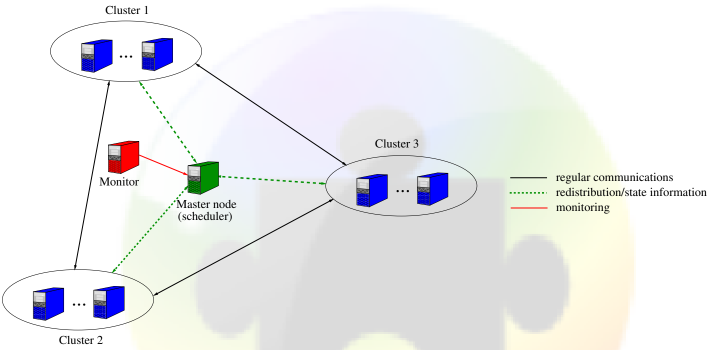


Figure: Experimental scheme: the *master* and the *workers*.

Master:

- Gather the results of the measurements.
- Call the redistribution algorithms when needed.

The *master* and the *workers*

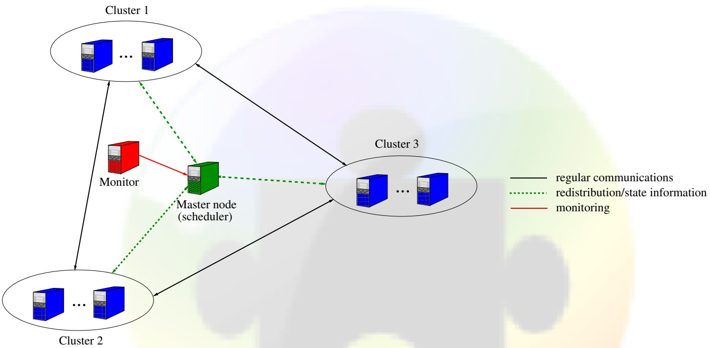


Figure: Experimental scheme: the *master* and the *workers*.

Monitor:

- Modify (using `wrekavoc`) the characteristics of the platform.

The *master* and the *workers*

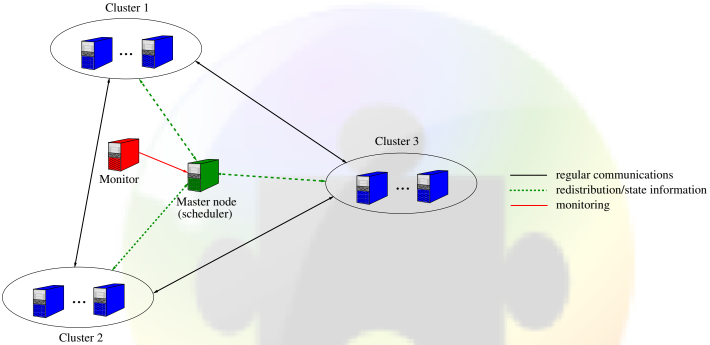


Figure: Experimental scheme: the *master* and the *workers*.

Slaves:

- Do all the computations and communications.
- Exchange data for redistribution according to the results of the master.

Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

Wrekavoc, in the center of both platforms

1. In our context, Wrekavoc is used to
   - ➥ control CPU and network capabilities;
   - ➥ of randomly chosen resources;
   - ➥ in order to study the behavior of the application.

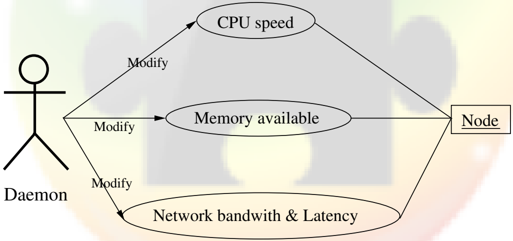


Figure: Wrekavoc in pictures

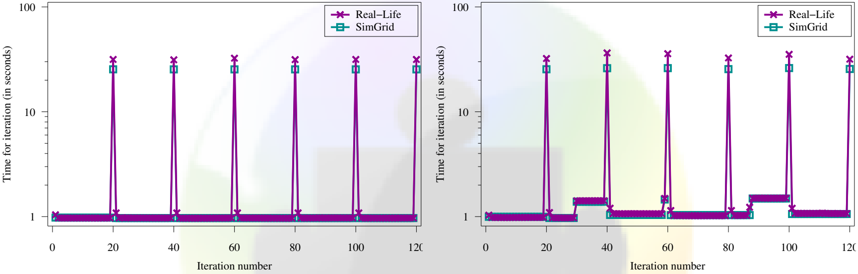
Grid'5000 vs SimGrid

1. Real and simulated execution:
   - Retrieve through measurements:
     - ☑ processor speed
     - ☑ network latency
     - ☑ inbound bandwidth
     - ☑ ...
2. Differences:
   - Real execution: the modification of the characteristics of the platform are done using `wrekavoc`,
   - Simulated execution: the modification of the characteristics of the platform is a built-in functionality of `SimGrid`.

Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

Grid'5000 vs SimGrid

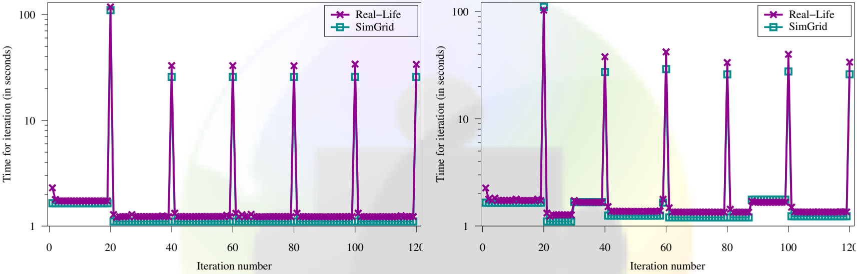


(a) No platform variation.

(b) With platform variation *(3 platform variations, once every 29 iterations).*

Figure: Time needed (in seconds) for each iteration on the real-life and the simulated platform: one site platform.

Grid'5000 vs SimGrid

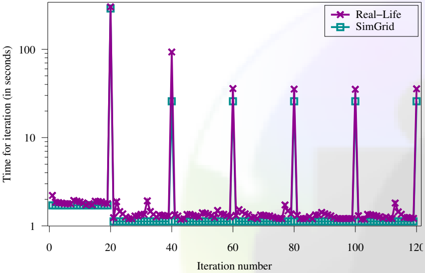


(a) No platform variation.

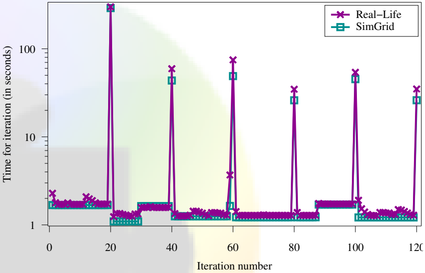
(b) With platform variation *(3 platform variations, once every 29 iterations).*

Figure: Time needed (in seconds) for each iteration on the real-life and the simulated platform: two sites platform.

Grid'5000 vs SimGrid



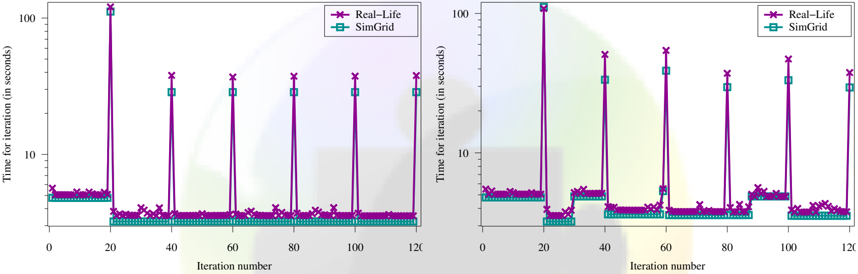
(a) No platform variation.



(b) With platform variation *(3 platform variations, once every 29 iterations).*

Figure: Time needed (in seconds) for each iteration on the real-life and the simulated platform: five sites platform.

Grid'5000 vs SimGrid



(a) No platform variation.

(b) With platform variation *(3 platform variations, once every 29 iterations).*

Figure: Time needed (in seconds) for each iteration on the real-life and the simulated platform: two sites platform. Each iteration is three time more costly than a regular one.

Plan of presentation

1. Framework
   - Data redistribution algorithms
   - Heat propagation
2. Real-life and simulation
   - Grid'5000 vs SimGrid
   - Wrekavoc
3. Experimental results
4. Conclusion

Conclusion

1. Two versions of the same application: the propagation of heat
   - Simulated implementation on top of `SimGrid`.
   - Real-life implementation running on the `Grid'5000` platform.
     - Using `wrekavoc` to control the characteristics of the platform.
   - Use the same platform characteristics over time in the two contexts.
2. The observed behavior for the simulated case is very close to that of a real execution.
3. A first step for validation of `SimGrid` in the context of complex applications.