Optimizing the execution of a parallel meteorology simulation code

Sonia Jerez, Juan Pedro Montávez, Domingo Giménez

Departamento de Física
University of Murcia, Spain
sonia.jerez@gmail.com, montavez@um.es, domingo@um.es
http://chubasco.inf.um.es

Departamento de Informática y Sistemas
http://dis.um.es/~domingo

International Workshop on Parallel and Distributed Scientific and Engineering Computing - IPDPS, Rome, May 29 2009
Contents

1. Motivation
2. Optimization scheme
3. The problem
4. Running conditions
5. Tools
6. Experiments
7. Conclusions
Use of parallel computing to solve computationally demanding problems ⇒ **Optimization**

- Adaptation of the routines to different architectures and optimization transparent to the user ⇒ **Self-optimization**
- Modelling the execution time ⇒ **Parametrization**
- Parallel and very complex scientific codes ⇒ **Black-box**
- Difficult modellization ⇒ **Empirical estimation**
Motivation

- Use of parallel computing to solve computationally demanding problems ⇒ **Optimization**

- Adaptation of the routines to different architectures and optimization transparent to the user ⇒ **Self-optimization**

- Modelling the execution time ⇒ **Parametrization**

- Parallel and very complex scientific codes ⇒ **Black-box**

- Difficult modellization ⇒ **Empirical estimation**
Motivation

- Use of parallel computing to solve computationally demanding problems ⇒ **Optimization**
- Adaptation of the routines to different architectures and optimization transparent to the user ⇒ **Self-optimization**
- Modelling the execution time ⇒ **Parametrization**
- Parallel and very complex scientific codes ⇒ **Black-box**
- Difficult modellization ⇒ **Empirical estimation**
Use of parallel computing to solve computationally demanding problems $\Rightarrow$ **Optimization**

Adaptation of the routines to different architectures and optimization transparent to the user $\Rightarrow$ **Self-optimization**

Modelling the execution time $\Rightarrow$ **Parametrization**

Parallel and very complex scientific codes $\Rightarrow$ **Black-box**

Difficult modellization $\Rightarrow$ **Empirical estimation**
Use of parallel computing to solve computationally demanding problems ⇒ Optimization

Adaptation of the routines to different architectures and optimization transparent to the user ⇒ Self-optimization

Modelling the execution time ⇒ Parametrization

Parallel and very complex scientific codes ⇒ Black-box

Difficult modellization ⇒ Empirical estimation
Optimization architecture

DESIGN
- routine design
- building cost function
  - obtaining value of system parameters on the actual system

INSTALLATION
- including system parameters in the cost function

EXECUTION
- obtaining the optimum values of the algorithmic parameters
- routine execution
Modifications on the optimization architecture

**DESIGN**
- Black-box designed
- Theoretical-empirical analysis

**INSTALLATION**
- Package OPTIM
  - Including system parameters in the cost function
  - Obtaining value of system parameters on the actual system

**EXECUTION**
- Package RUN_OPTIM
  - Obtaining the optimum values of the algorithmic parameters
  - Routine execution

---

Motivation | Optimization scheme | The problem | Running conditions | Tools | Experiments | Conclusions
Regional Climate Models

- Solve limited areas and permit higher spatial resolution with a computational cost lower than that of general models
- MM5 developed at the Pennsylvania State University and the National Center for Atmospheric Research
  Parallel versions with OpenMP and MPI
- Regional Atmospheric Modelling Group at the University of Murcia applies them to
  - Global warming impact at regional scales
  - Evaluation of renewable energy resources
  - Air quality evaluation models
Regional Climate Models

- Solve limited areas and permit higher spatial resolution with a computational cost lower than that of general models
- **MM5** developed at the Pennsylvania State University and the National Center for Atmospheric Research
  Parallel versions with OpenMP and MPI
- Regional Atmospheric Modelling Group at the University of Murcia applies them to
  - Global warming impact at regional scales
  - Evaluation of renewable energy resources
  - Air quality evaluation models
Regional Climate Models

- Solve limited areas and permit higher spatial resolution with a computational cost lower than that of general models
- **MM5** developed at the Pennsylvania State University and the National Center for Atmospheric Research
  Parallel versions with OpenMP and MPI
- Regional Atmospheric Modelling Group at the University of Murcia applies them to
  - Global warming impact at regional scales
  - Evaluation of renewable energy resources
  - Air quality evaluation models
Regional Climate Models

- Solve limited areas and permit higher spatial resolution with a computational cost lower than that of general models
- **MM5** developed at the Pennsylvania State University and the National Center for Atmospheric Research
  Parallel versions with OpenMP and MPI
- Regional Atmospheric Modelling Group at the University of Murcia applies them to
  - Global warming impact at regional scales
  - Evaluation of renewable energy resources
  - Air quality evaluation models
Applications at the Regional Atmospheric Modelling Group

Two two-way nested tri-dimensional domains D1 and D2. Determined by convergence laws.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Temp. Step</th>
<th>Num. Steps D2/D1</th>
<th>Sigma Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>30 km</td>
<td>90 sec</td>
<td>27</td>
</tr>
<tr>
<td>D2</td>
<td>10 km</td>
<td>270 sec</td>
<td>24</td>
</tr>
</tbody>
</table>

- DOI: 10.3390/xxxxxx
- Temp: 27°C
- Humidity: 70%
- Pressure: 1013 mbar
- Wind: 5 m/s, direction: N
- Precipitation: 10 mm/day

[Map of Domains D1 and D2]
Related applications

- The typical Jacobi iteration
- Applications in simulation of petrol extraction
- Hydrodynamic simulations, maritime contamination

In the methods with two types of iterations, the influence of the number of iterations of each type on the convergence speed should be studied
Related applications

- The typical Jacobi iteration
- Applications in simulation of petrol extraction
- Hydrodynamic simulations, maritime contamination
- In the methods with two types of iterations, the influence of the number of iterations of each type on the convergence speed should be studied
Running conditions

After the simulation of a period of fixed length (spin-up period, $T_s$) the influence of the initial condition is discarded. The value of $T_s$ depends on each experiment.

- **Time parallelization:**
  Divide the period $P$ in $N_t$ subperiods and simulate each subperiod with the spin-up time $T_s$:

  $$T = \left( \frac{P}{N_t} + T_s \right) t$$

  where $t$ is the cost of the simulation of a unity-length period with the chosen configuration

  $P = 4$ years
  
  $N = 4$ cores

  $N_t = N = 4$
Running conditions

After the simulation of a period of fixed length (spin-up period, $T_s$) the influence of the initial condition is discarded. The value of $T_s$ depends on each experiment.

- **Time parallelization:**
  Divide the period $P$ in $N_t$ subperiods and simulate each subperiod with the spin-up time $T_s$:

\[
T = \left( \frac{P}{N_t} + T_s \right) t
\]

where $t$ is the cost of the simulation of a unity-length period with the chosen configuration

\[
\begin{align*}
P &= 4 \text{ years} \\
N &= 4 \text{ cores}
\end{align*}
\]

\[
N_t = N = 4
\]
**Spatial parallelization:** Using the PARALLEL CODE that divides the spatial domain, and each portion is solved simultaneously in each core involved in the execution. Use $N_p = N_x N_y$ cores for each simulation. The total number of cores is $N = N_t N_p$. The cost of a basic operation depends on the parameters: $t = f (N_t, N_x, N_y)$ and mesh configuration.
Installation: OPTIM

- A short period of time is simulated for all the possible combinations of $N_t$ with $N_p$
- with a limit: $N_t N_p \leq 2N$
- for some trial domains
- and different mesh shapes: combinations of $N_x$ and $N_y$
- Indicate in the installation:
  - Where package MM5 is
  - The number of available processors
  - Compilation options
  - The manager could decide modify some of the default parameters
A short period of time is simulated for all the possible combinations of $N_t$ with $N_p$

with a limit: $N_t N_p \leq 2N$

for some trial domains

and different mesh shapes: combinations of $N_x$ and $N_y$

Indicate in the installation:

- Where package MM5 is
- The number of available processors
- Compilation options
- The manager could decide modify some of the default parameters
Installation: OPTIM

- A short period of time is simulated for all the possible combinations of $N_t$ with $N_p$
- with a limit: $N_t N_p \leq 2N$
- for some trial domains
- and different mesh shapes: combinations of $N_x$ and $N_y$
- Indicate in the installation:
  - Where package MM5 is
  - The number of available processors
  - Compilation options
  - The manager could decide modify some of the default parameters
Installation: OPTIM

- A short period of time is simulated for all the possible combinations of $N_t$ with $N_p$
- with a limit: $N_t N_p \leq 2N$
- for some trial domains
- and different mesh shapes: combinations of $N_x$ and $N_y$

Indicate in the installation:
- Where package MM5 is
- The number of available processors
- Compilation options
- The manager could decide modify some of the default parameters
Installation: OPTIM

- A short period of time is simulated for all the possible combinations of $N_t$ with $N_p$
- with a limit: $N_t N_p \leq 2N$
- for some trial domains
- and different mesh shapes: combinations of $N_x$ and $N_y$

Indicate in the installation:
- Where package MM5 is
- The number of available processors
- Compilation options
- The manager could decide modify some of the default parameters
Execution: RUNOPTIM

- Select at running time the values of $N_t$, $N_x$ and $N_y$
- Taking into consideration the size and characteristics of the problem to be solved
- With the values $t = f(N_t, N_x, N_y)$ estimated at installation time by OPTIM for domains close to the current domain
- To update the information generated by OPTIM for the problem:
  - Overhead
  - Possibly the estimation adjusts better to the problem characteristics
Execution: RUNOPTIM

- Select at running time the values of $N_t$, $N_x$ and $N_y$
- Taking into consideration the size and characteristics of the problem to be solved
- With the values $t = f(N_t, N_x, N_y)$ estimated at installation time by OPTIM for domains close to the current domain
- To update the information generated by OPTIM for the problem:
  - Overhead
  - Possibly the estimation adjusts better to the problem characteristics
Execution: RUNOPTIM

- Select at running time the values of $N_t$, $N_x$ and $N_y$
- Taking into consideration the size and characteristics of the problem to be solved
- With the values $t = f(N_t, N_x, N_y)$ estimated at installation time by OPTIM for domains close to the current domain
- To update the information generated by OPTIM for the problem:
  - Overhead
  - Possibly the estimation adjusts better to the problem characteristics
Execution: RUNOPTIM

- Select at running time the values of $N_t$, $N_x$ and $N_y$
- Taking into consideration the size and characteristics of the problem to be solved
- With the values $t = f(N_t, N_x, N_y)$ estimated at installation time by OPTIM for domains close to the current domain
- To update the information generated by OPTIM for the problem:
  - Overhead
  - Possibly the estimation adjusts better to the problem characteristics
RUNOPTIM: utilization options

- **Use information of OPTIM:**
  - Select trial domains with the shapes closest to the current domain
  - Select trial domain with the size closest to the current one
  - Use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$ and $N_y$
  - **INSTAL:** selects the values which gives lowest modelled time
  - **INS+EXE:** repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- **Not using information from OPTIM:**
  - **DEFAUL:** uses default parameters
  - **EXECUT:** repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
RUNOPTIM: utilization options

- Use information of OPTIM:
  - Select trial domains with the shapes closest to the current domain
  - Select trial domain with the size closest to the current one
  - Use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$ and $N_y$
  - **INSTAL**: selects the values which gives lowest modelled time
  - **INS+EXE**: repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- Not using information from OPTIM:
  - **DEFAUL**: uses default parameters
  - **EXECUT**: repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
RUNOPTIM: utilization options

- **Use information of OPTIM:**
  - Select trial domains with the shapes closest to the current domain
  - Select trial domain with the size closest to the current one
  - Use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$, and $N_y$
  - **INSTAL:** selects the values which gives lowest modelled time
  - **INS+EXE:** repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- **Not using information from OPTIM:**
  - **DEFAUL:** uses default parameters
  - **EXECUT:** repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
RUNOPTIM: utilization options

- **Use information of OPTIM:**
  - Select trial domains with the shapes closest to the current domain
  - select trial domain with the size closest to the current one
  - use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$ and $N_y$
  - **INSTAL:** selects the values which gives lowest modelled time
  - **INS+EXE:** repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- **Not using information from OPTIM:**
  - **DEFAUL:** uses default parameters
  - **EXECUT:** repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
RUNOPTIM: utilization options

- Use information of OPTIM:
  - Select trial domains with the shapes closest to the current domain
  - select trial domain with the size closest to the current one
  - use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$ and $N_y$
  - **INSTAL**: selects the values which gives lowest modelled time
  - **INS+EXE**: repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- Not using information from OPTIM:
  - **DEFAUL**: uses default parameters
  - **EXECUT**: repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
RUNOPTIM: utilization options

- Use information of OPTIM:
  - Select trial domains with the shapes closest to the current domain
  - select trial domain with the size closest to the current one
  - use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$ and $N_y$
  - **INSTAL**: selects the values which gives lowest modelled time
  - **INS+EXE**: repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- Not using information from OPTIM:
  - **DEFUAL**: uses default parameters
  - **EXECUT**: repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
RUNOPTIM: utilization options

- Use information of OPTIM:
  - Select trial domains with the shapes closest to the current domain
  - Select trial domain with the size closest to the current one
  - Use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$, and $N_y$
  - **INSTAL**: selects the values which give lowest modelled time
  - **INS+EXE**: repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- Not using information from OPTIM:
  - **DEFAUL**: uses default parameters
  - **EXECUT**: repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
RUNOPTIM: utilization options

- **Use information of OPTIM:**
  - Select trial domains with the shapes closest to the current domain
  - select trial domain with the size closest to the current one
  - use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$, and $N_y$
  - **INSTAL**: selects the values which gives lowest modelled time
  - **INS+EXE**: repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- **Not using information from OPTIM:**
  - **DEFAUL**: uses default parameters
  - **EXECUT**: repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
RUNOPTIM: utilization options

- **Use information of OPTIM:**
  - Select trial domains with the shapes closest to the current domain
  - Select trial domain with the size closest to the current one
  - Use the value of $t$ obtained by OPTIM for the different combinations of $N_t$, $N_x$, and $N_y$
  - **INSTAL:** selects the values which gives lowest modelled time
  - **INS+EXE:** repeats the experiments of OPTIM for the current problem for the parameter combinations which provide lowest modelled time

- **Not using information from OPTIM:**
  - **DEFAUL:** uses default parameters
  - **EXECUT:** repeats OPTIM execution for the current domain, and selects the parameters which give the lowest estimated time
**Systems**

- **RAYO**: 16 nodes Intel Quad Q6600
  - Experiments
  - Better results with $N_t$ for various nodes $\implies$ experiments in one node

- **HIPATIA**: 2 nodes with four Xeon X7350 Quad + 14 nodes with two Xeon E5462 Quad
  - Experiments in node with eight cores
**Installation**

Installation in cluster RAYO: simulation period of one year, 380 simulations, 33 hours (HIPATIA 21 hours)

Optimum values for different domain sizes and spin-up periods:

<table>
<thead>
<tr>
<th>domain points E-W×N-S</th>
<th>10 days $N_t \ N_x \ N_y$</th>
<th>1 month $N_t \ N_x \ N_y$</th>
<th>3 months $N_t \ N_x \ N_y$</th>
<th>6 months $N_t \ N_x \ N_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30×30</td>
<td>7 1 1</td>
<td>4 1 1</td>
<td>2 2 1</td>
<td>2 2 2</td>
</tr>
<tr>
<td>50×50</td>
<td>7 1 1</td>
<td>7 1 1</td>
<td>1 4 1</td>
<td>1 4 1</td>
</tr>
<tr>
<td>80×80</td>
<td>3 2 1</td>
<td>3 2 1</td>
<td>1 4 1</td>
<td>1 4 1</td>
</tr>
<tr>
<td>120×120</td>
<td>3 2 1</td>
<td>3 2 1</td>
<td>1 4 1</td>
<td>1 4 1</td>
</tr>
<tr>
<td>60×30</td>
<td>5 1 1</td>
<td>5 1 1</td>
<td>1 4 1</td>
<td>1 4 1</td>
</tr>
<tr>
<td>90×50</td>
<td>5 1 1</td>
<td>3 2 1</td>
<td>1 4 1</td>
<td>1 4 1</td>
</tr>
<tr>
<td>130×80</td>
<td>6 1 1</td>
<td>2 3 1</td>
<td>1 4 1</td>
<td>1 4 1</td>
</tr>
<tr>
<td>30×60</td>
<td>5 1 1</td>
<td>5 1 1</td>
<td>2 2 1</td>
<td>1 2 2</td>
</tr>
<tr>
<td>50×90</td>
<td>5 1 1</td>
<td>5 1 1</td>
<td>1 4 1</td>
<td>1 4 1</td>
</tr>
<tr>
<td>80×130</td>
<td>3 2 1</td>
<td>2 3 1</td>
<td>1 4 1</td>
<td>1 4 1</td>
</tr>
</tbody>
</table>

Conclusions not intuitive for non-expert users $\Rightarrow$ automatic selection of parameters
Cluster RAYO, execution time in hours

Domain 1
8 days
\(T_s = 2\) days

Domain 2
12 months
\(T_s = 3\) months
Simulation 12 months, $T_s = 3$ months

RAYO
Domain 1
exe. days

HIPATIA
Domain 2
exe. hours
Conclusions

- Methodology to optimize the execution time of a meteorological simulation code
- Code very complex, treated as black-box
- Installation and execution tools
- A reduction of the execution time of 25% - 40%
- Need for more experiments, mainly in systems and problems for which the use of the complete cluster could contribute to reduce the execution time
- Application to other simulation codes (hydrodynamic simulations...)
- In some of these codes, study of the influence on the convergence of the number of iterations of each type
Conclusions

- Methodology to optimize the execution time of a meteorological simulation code
- Code very complex, treated as black-box
- Installation and execution tools
- A reduction of the execution time of 25% - 40%
- Need for more experiments, mainly in systems and problems for which the use of the complete cluster could contribute to reduce the execution time
- Application to other simulation codes (hydrodynamic simulations...)
- In some of these codes, study of the influence on the convergence of the number of iterations of each type
Optimizing the execution of a parallel meteorology simulation code

Sonia Jerez, Juan Pedro Montávez, Domingo Giménez

Departamento de Física
University of Murcia, Spain
sonia.jerez@gmail.com, montavez@um.es, domingo@um.es
http://chubasco.inf.um.es

Departamento de Informática y Sistemas
http://dis.um.es/~domingo

International Workshop on Parallel and Distributed Scientific and Engineering Computing - IPDPS, Rome, May 29 2009