ELASTIC: Dynamic Tuning for Large-Scale Parallel Applications

Toni Espinosa
Andrea Martínez, Anna Sikora, Eduardo César and Joan Sorribes

Universitat Autònoma de Barcelona
Computer Architecture and Operating Systems Departament
Outline

① Motivation.

② Scalable Dynamic Tuning.

③ ELASTIC.

④ Experimental Evaluation.

⑤ Conclusions and Future Work.
Motivation

Centralised Architecture of Tuning Tools

- Elimination of a single centralised control point.
- Distribution of the analysis and tuning process, remaining effective.
Outline

① Motivation.

② Scalable Dynamic Tuning.

③ ELASTIC.

④ Experimental Evaluation.

⑤ Conclusions and Future Work.
Hierarchical Tuning Network

➢ *Decompose.*

- A base level of **analysis and tuning modules** (ATM) that controls disjoint **domains** of application tasks.

Local performance improvements are achieved.

…and how to obtain global performance improvements?
Hierarchical Tuning Network

Abstract.

- The abstraction mechanism is carried out by the ATMs.

...representing the tasks of the virtual parallel application
Hierarchical Tuning Network

- **Decompose.**
- **Abstract.**

The actuation of each ATM of the network gives a hierarchical distribution of the analysis and tuning process located at the high level.
Abstraction Mechanism
Knowledge in the Tuning Network

Performance Model
- Monitoring Points
- Performance Expressions
- Tuning Points, Actions and Synchronisation Method

Abstraction Model
- How to translate a monitoring order
- How to translate a tuning order
- How to create a new event
  - How to decompose the real or virtual parallel application
Outline

① Motivation.

② Scalable Dynamic Tuning.

③ ELASTIC.

④ Experimental Evaluation.

⑤ Conclusions and Future Work.
ELASTIC

- Prototype implementation in C++.
- For MPI parallel applications.
- Target systems: UNIX based supercomputers.

![Diagram of ELASTIC process]

- Source Code
- Executable
- Application Memory
- Monitoring
- Analysis
- Tuning

- Dynamic instrumentation
- Event tracing
- Performance and abstraction models

Execution time
ELASTIC: Architecture

The tuning network topology can be configured to accommodate the size of the parallel application and the complexity of the tuning strategy being employed.
ELASTIC Package

Set of code and configurations that implements the performance and abstraction model.

**Performance Model**
- Monitoring Points
- Performance Expressions
- Tuning Points, Actions and Synchronisation Method

**Abstraction Model**
- How to translate a monitoring order
- How to translate a tuning order
- How to create a new event
- How to decompose the real or virtual parallel application

**TUNING AND ABSTRACTION API**
- `vector<Monitoring Order> PerformanceEvaluator::InitialMonitoringOrders()`
- `bool PerformanceEvaluator::NewEvent(Event *e)`
- `vector<Order> PerformanceEvaluator::EvaluatePerformance()`
- `vector<Monitoring Order> InstrumentationOrderTranslator::TranslateMonitoringOrder(MonitoringOrder *mo)`
- `vector<Tuning Order> InstrumentationOrderTranslator::TranslateTuningOrder(TuningOrder *to)`
- `bool EventCreator::NewEvent(Event *e)`
- `vector<Event> EventCreator::CreateEvent()`
ELASTIC Package

Plug-in Architecture

• Codification of the ELASTIC Package based on subclassing Abstractor-ATM components.

This plugin architecture converts ELASTIC into a general purpose tuning tool and gives it the flexibility to tackle a wide range of performance problems.
Outline

① Motivation.

② Scalable Dynamic Tuning.

③ ELASTIC.

④ Experimental Evaluation.

⑤ Conclusions and Future Work.
Experimental Evaluation

The evaluation consists of

Executing a parallel application which presents a specific performance problem and using ELASTIC to dynamically detect and resolve the problem.

Execution Environment: **Supercomputer SuperMUC at LRZ.**
- 9400 compute nodes (155656 cores).
- Each node has 2 8-core 2.7 GHz Intel Xeon processors and 32GB main memory.
- SuSe Linux.
Experimental Evaluation

Results

SYNTHETIC APPLICATION

Execution time (s)

Number of tasks

Original application
Application tuned by ELASTIC

<table>
<thead>
<tr>
<th>Number of tasks</th>
<th>Original application</th>
<th>Application tuned by ELASTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>53.4%</td>
<td>25.8%</td>
</tr>
<tr>
<td>1024</td>
<td>50%</td>
<td>29.4%</td>
</tr>
<tr>
<td>2304</td>
<td>46.8%</td>
<td>40.6%</td>
</tr>
<tr>
<td>4096</td>
<td></td>
<td>50%</td>
</tr>
<tr>
<td>9216</td>
<td>29.4%</td>
<td>46.8%</td>
</tr>
<tr>
<td>16384</td>
<td></td>
<td>50%</td>
</tr>
</tbody>
</table>
Experimental Evaluation

Results

AGENT-BASED APPLICATION

![Bar chart showing execution time for different numbers of tasks.]

- Original application
- Application tuned by ELASTIC

Execution time (s)

Execution time increases with the number of tasks. The application tuned by ELASTIC shows a decrease in execution time compared to the original application.

Number of tasks

- 256 tasks: 20.8%
- 512 tasks: 28.3%
- 1024 tasks: 30.3%
- 2048 tasks: 31.1%
Outline

① Motivation.

② Scalable Dynamic Tuning.

③ ELASTIC.

④ Experimental Evaluation.

⑤ Conclusions and Future Work.
Conclusions

✓ The distribution of the dynamic tuning process through a hierarchical tuning network of analysis modules has been defined.

✓ ELASTIC, a tool that implements the proposed design, has been developed.
  ➢ It offers dynamic tuning through dynamic monitoring, automatic performance analysis and dynamic modifications.
  ➢ It presents an adaptable topology and a plugin architecture.

✓ The encouraging results obtained from the experimental evaluation using ELASTIC show that our approach is effective for large-scale dynamic tuning.
Future Work

✓ Creation of **general** ELASTIC Packages which solve a given performance problem.
  
  ➢ It would be required a **small adaptation** to applied them to specific parallel applications.

✓ **Combine** our approach with the one implemented under the AutoTune project.
ELASTIC: Dynamic Tuning for Large-Scale Parallel Applications

Thank you

Toni Espinosa, Andrea Martinez, Anna Sikora, Eduardo César and Joan Sorribes
Universitat Autònoma de Barcelona
Computer Architecture and Operating Systems Departament
Tuning Network Topology

**HOW DO WE SELECT A TUNING NETWORK TOPOLOGY?**

Parallel Application of N tasks

The structure of the topology will depend on the number of levels in the hierarchy and the number of Abstractor-ATM pairs in each level.

The use of tuning networks composed of the minimum number of non-saturated Abstractor-ATM pairs.

The maximum domain size that an Abstractor-ATM pair can manage without becoming saturated.
Modelling an analysis and tuning process

\[ N \times T_m \times E_a + T_a(N) + \frac{E_a}{E_c} \times T_c + T_t \times f_{rp} \]

Event Batch

\[ N \times T_m \ldots N \times T_m \ldots N \times T_m \ldots \]

\[ T_a(N) \]

Instrumentation orders from the parent level

Events to the parent level

Instrumentation orders to the immediate children

ATM

Event Creator

Abstractor
Calculating the number of tasks that an analysis module can manage without becoming saturated

\[ f_{\text{analysis}} = \frac{f_e}{E_a} \]

Period_{analysis} + Period_{analysis} = 

N * T_m * E_a + T_a(N) + E_a * T = 

Parallel Application of 256 tasks

\begin{align*}
T_a & \quad E_a \quad f_a \\
T_m & \quad E_a \quad f_e \\
T_c & \quad E_c \quad f_{rp} \\
T_t & 
\end{align*}
Experimental Evaluation

Application and performance problem

- **Logical layout:** 2D grid.

- **Iteration pattern:**
  - Computation.
  - Communication.

- **Load imbalance:** hotspots of additional workload were introduced into the application at runtime.

### Scenario 1  10%

### Scenario 2  21%
Experimental Evaluation
ELASTIC Package to balance the work units

Performance Model

Abstraction Model

Performance Expressions

**LOAD BALANCE ALGORITHM**

*Constraint:* the work units only can be moved between neighbouring tasks

**Input:** work units.

**Output:** work units to send.

Measurement Points

- **Per task**
  - work units
  - iteration id
  - task id

**Place**

- work() (stub)

Tuning Points, Action and Synchronisation

- **Points:** [send_north, send_south, send_east, send_west]
- **Action:** set the value of these variables.
- **Synchronisation:** at the beginning of the migration phase.
- **Migration function**
Experimental Evaluation
ELASTIC Package to balance the work units

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: communication pattern between tasks

Real/Virtual Application

Tuning Order Translation

Tuning Network

Monitoring Order Translation & Event Creation

Tuning Network

Monitoring Order

Events

\[ \sum_{\text{work units}} \]

\([\text{send\_south}, 60]\]

\([\text{send\_south}, 20]\)

\([\text{send\_south}, 20]\)

\([\text{send\_south}, 20]\)
Experimental Evaluation

Experimentation Plan

For the two scenarios

<table>
<thead>
<tr>
<th>Application Tasks</th>
<th>Level 0</th>
<th>Level 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of ATMs</td>
<td>Number of ATMs</td>
</tr>
<tr>
<td>256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2304</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4096</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16384</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 100 iterations.
- 20 work units per task.
- The additional load is proportional to the size of the parallel application.
Experimental Evaluation

Results

**SCENARIO 1**

Load State

4096 tasks parallel application (64x64 grid)

Original application

Virtual application
Experimental Evaluation

Results

SCENARIO 1

---

**Execution time (s)**

<table>
<thead>
<tr>
<th>Number of tasks</th>
<th>Original application</th>
<th>Application tuned by ELASTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>65.1%</td>
<td>22.9%</td>
</tr>
<tr>
<td>1024</td>
<td>63.1%</td>
<td></td>
</tr>
<tr>
<td>2304</td>
<td>54.6%</td>
<td></td>
</tr>
<tr>
<td>4096</td>
<td>50.1%</td>
<td></td>
</tr>
<tr>
<td>9216</td>
<td>33.1%</td>
<td></td>
</tr>
<tr>
<td>16384</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experimental Evaluation

SCENARIO 2

Iteration Time

Results

256 Tasks Synthetic Application

1024 Tasks Synthetic Application

2304 Tasks Synthetic Application

256 Tasks Synthetic Application

1024 Tasks Synthetic Application

2304 Tasks Synthetic Application

4096 Tasks Synthetic Application

9216 Tasks Synthetic Application

16384 Tasks Synthetic Application
Experimental Evaluation

Results

SCENARIO 2

Number of tasks

Execution time (s)

Original application
Application tuned by ELASTIC

53.4% 50% 46.8% 40.6% 29.4% 25.8%
Experimental Evaluation

- Synthetic Parallel Application
- Real Agent-based Parallel Application

- Application and performance problem.
- ELASTIC Package developed.
- Experimentation plan.
- Results.
Experimental Evaluation

Application and performance problem

- Large-scale agent-based simulation.
- Simulates an epidemic model.
- Communication pattern: any-to-any

- Load imbalance problem due to the dynamic behaviour of the agents:
  - Births and death.
  - Time required to process an agent is not uniform.
Performance Model

Abstraction Model

Performance Expressions

LOAD BALANCE ALGORITHM

Marquez et al. 2013

Migrations can be between any two tasks in the analysis and tuning domain

Input: #agents and computation time.
Output: #agents to send to each task.

Measurement Points

Per task
- time
- # agent
- iteration id
- task id

Place
- phase_4()

Tuning Points, Action and Synchronisation
- Points: [intradomain_migrate[], interdomain_migrate[]]
- Action: set the value of these variables.
- Synchronisation: at the beginning of the migration phase.
- Migration functions.

Experimental Evaluation
ELASTIC Package to balance the computation time
Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time

Performance Model

Abstraction Model

Decomposition Scheme

Constraint: domains with the same size

Real/Virtual Application

Monitoring Order Translation & Event Creation

Tuning Network

Constraint: domains with the same size

Tuning Order Translation

[intradomain, 80]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]
[interdomain, 20]

Tuning Network

Experimental Evaluation
ELASTIC Package to balance the computation time
Experimental Evaluation
Experimentation Plan

4 SIMULATION SCENARIOS

- Scale the number of agents.
- Scale the simulation space.

Domain size = 512

<table>
<thead>
<tr>
<th>Number of Application Tasks</th>
<th>Number of Agents</th>
<th>Simulated Space Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>37,500</td>
<td>1020×1020</td>
</tr>
<tr>
<td>512</td>
<td>75,000</td>
<td>1440×1440</td>
</tr>
<tr>
<td>1,024</td>
<td>150,000</td>
<td>1800×1800</td>
</tr>
<tr>
<td>2,048</td>
<td>300,000</td>
<td>2240×2240</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Application Tasks</th>
<th>Level 0</th>
<th>Level 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of ATMs</td>
<td>Number of ATMs</td>
</tr>
<tr>
<td>256</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>512</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>1,024</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2,048</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
Experimental Evaluation

Results

256 tasks

512 tasks

1024 tasks

2048 tasks

Computation times - 37500 agents on 256 tasks
Agent-based application alone
Agent-based application with ELASTIC
Ideal balanced time

Computation times - 75000 agents on 512 tasks
Agent-based application alone
Agent-based application with ELASTIC
Ideal balanced time

Computation times - 150000 agents on 1024 tasks
Agent-based application alone
Agent-based application with ELASTIC
Ideal balanced time

Computation times - 300000 agents on 2048 tasks
Agent-based application alone
Agent-based application with ELASTIC
Ideal balanced time
Experimental Evaluation

Results

![Bar chart showing execution time and number of tasks for Original application and Application tuned by ELASTIC.]