POSTER CONTACTS:
Renato Miceli: renato.miceli@ichec.ie
Gilles Civario: gilles.civario@ichec.ie



# AutoTune: Automatic Online Code Tuning

**Renato Miceli[1], Gilles Civario[1], François Bodin[2]**

[1] Irish Centre for High-End Computing, Trinity Technology & Enterprise Campus, Grand Canal Quay, Dublin 2, Ireland
[2] CAPS Entreprise, Immeuble CAP Nord, Bât A, 4 Allée Marie Berhaut, 35000 Rennes, France
renato.miceli@ichec.ie, gilles.civario@ichec.ie, francois.bodin@caps-entreprise.com

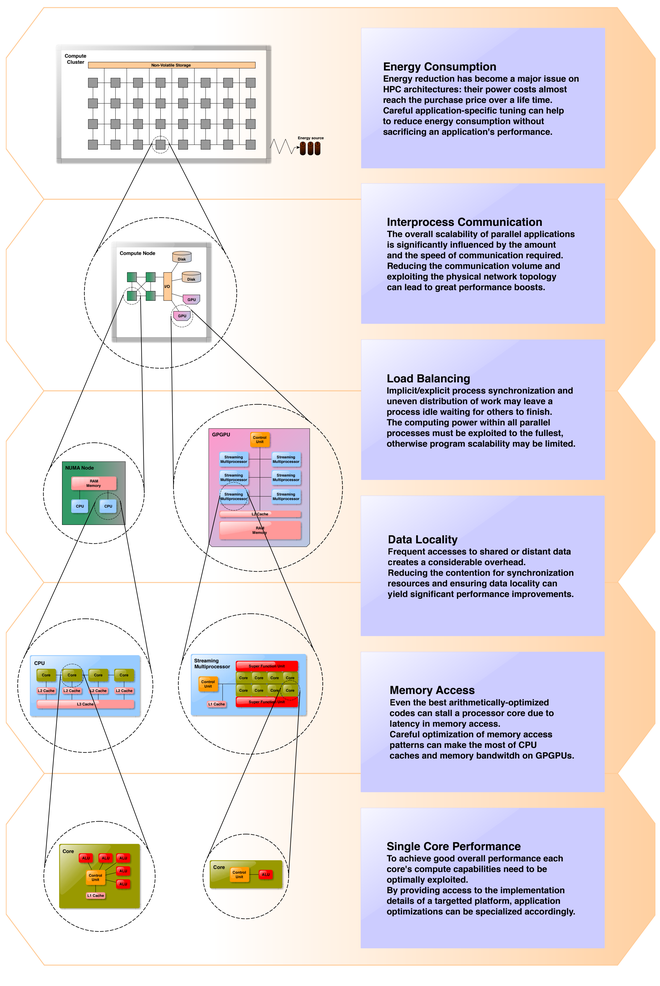
## 1. Abstract

Performance analysis and tuning is an important step in programming multicore and manycore architectures. There are several tools to help developers analyze application performance; still, no tool provides recommendations about how to tune the code.
AutoTune will extend Periscope, an automatic online and distributed performance analysis tool developed by Technische Universität München, with plugins for performance and energy efficiency tuning. The resulting Periscope Tuning Framework will be able to tune serial and parallel codes with and without GPU kernels; in addition, it will return tuning recommendations that can be integrated into the production version of the code. The whole tuning process, consisting of both automatic performance analysis and automatic tuning, will be executed online, i.e. during a single run of the application.

## 2. Motivation

### The Challenge of Programming Parallel Architectures

**The shift to multi- and many-core architectures made it more complex to develop hardware-optimized applications. A number of performance analysis tools exist to support the application tuning process, but none of them provide recommendations about how to tune the code.**




**Energy Consumption**
Energy reduction has become a major issue on HPC architectures: their power costs almost reach the purchase price over a life time. Careful application-specific tuning can help to reduce energy consumption without sacrificing an application's performance.

**Interprocess Communication**
The overall scalability of parallel applications is significantly influenced by the amount and the speed of communication required. Reducing the communication volume and exploiting the physical network topology can lead to great performance boosts.

**Load Balancing**
Implicit/explicit process synchronization and uneven distribution of work may leave a process idle waiting for others to finish. The computing power within all parallel processes must be exploited to the fullest, otherwise program scalability may be limited.

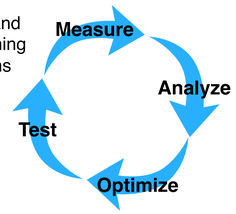
**Data Locality**
Frequent accesses to shared or distant data creates a considerable overhead. Reducing the contention for synchronization resources and ensuring data locality can yield significant performance improvements.

**Memory Access**
Even the best arithmetically-optimized codes can stall a processor core due to latency in memory access. Careful optimization of memory access patterns can make the most of CPU caches and memory bandwidth on GPGPUs.

**Single Core Performance**
To achieve good overall performance each core's compute capabilities need to be optimally exploited. By providing access to the implementation details of a targetted platform, application optimizations can be specialized accordingly.

## 3. Project Goals

The AutoTune Project's goal is to close the gap in the application tuning process and simplify the development of efficient parallel programs. It focuses on automatic tuning for multicore- and manycore-based parallel systems, ranging from desktop systems with and without GPGPUs to petascale and future exascale HPC architectures.
To achieve this objective, AutoTune aims at developing the Periscope Tuning Framework (PTF), the first framework to combine and automate both analysis and tuning into a single tool. AutoTune's PTF will…



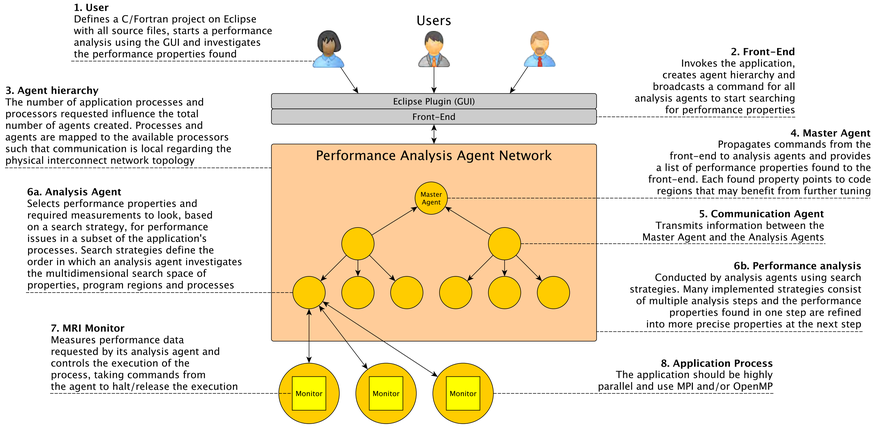

- Identify tuning alternatives based on codified expert knowledge.
- Evaluate the alternatives online (i.e. within the same application execution), reducing the overall search time for a tuned version.
- Produce a report on how to improve the code, which can be manually or automatically applied.

**Common performance analyzers only hint at where to tune. AutoTune's PTF will also tune the code for you!**

## 4. Periscope: the basis for AutoTune

### Architecture of Periscope

Developed by Technische Universität München, Periscope is an automatic performance analysis tool for highly parallel applications written in MPI and/or OpenMP. Periscope performs program analysis while the application is executing (online) using a set of analysis agents in a distributed fashion. It is available for all major parallel architectures and has production-level quality. Its scalability has been demonstrated on large BlueGene systems by analyzing applications running on up to 65,000 cores.




## 5. AutoTune's Tuning Framework

### a. Which Tuning Plugins?

- GPU programming with HMPP and OpenCL
- Single-core performance tuning
- MPI tuning
- Energy efficiency tuning

### b. How they will be implemented?

- **Master Agent:** responsible for implementing the overall tuning strategy
- **Analysis Agents:** may implement portions of the tuning plugins
- **MRI Monitor:** measures energy consumed and monitors the GPU infrastructure; may implement region-specific tuning actions (e.g. changing the clock frequency for a specific program region)
- **C/C++ and Fortran instrumenter:** extended for parallel pattern support and HMPP and OpenCL codes

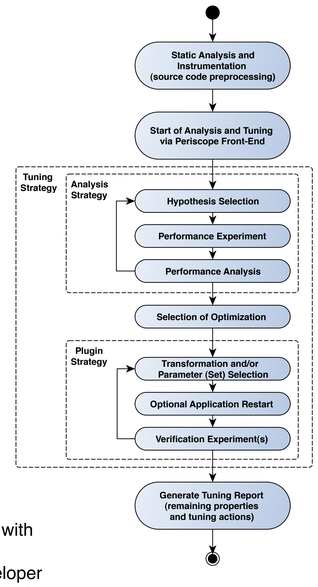
### c. Using which techniques?

- Expert knowledge
- Iterative search
- Machine learning
- Model-guided empirical optimization

PTF will take a program written in MPI/OpenMP with/without kernels for GPGPUs in HMPP or OpenCL and will automatically tune it with respect to performance and energy usage.
PTF will generate a tuning report such that a developer can integrate the tuning recommendations for production runs.

### PTF Tuning Control Flow

**AutoTune will develop the Periscope Tuning Framework (PTF) extending Periscope. It will follow Periscope's main principles, i.e. the use of formalized expert knowledge in form of properties and strategies, automatic execution, online search based on program phases, and distributed processing. Periscope will be extended by a number of online and semi-online tuning plugins responsible for searching for a tuned code version.**




## 6. Goal Validation

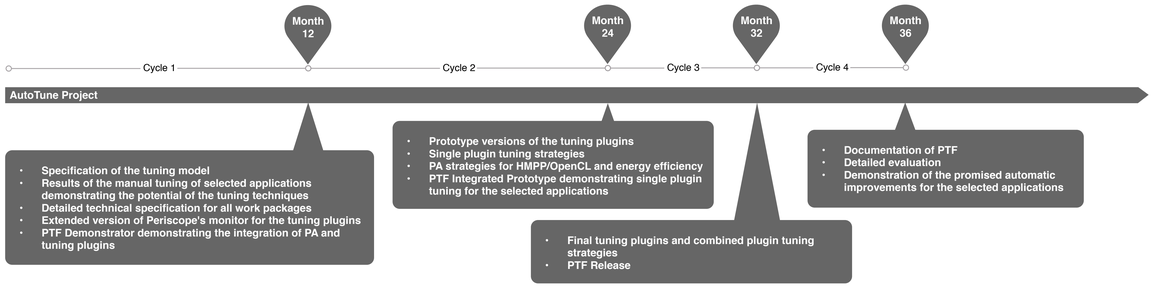
Achievement of the goals will be measured and evaluated at the project's end:

1. Applications that can benefit from the tuning techniques will be selected and manually tuned during the course of the project.
2. At the end of the project PTF will be run over the same applications.

The improvements achieved and the required effort for both the manual and the automatic tuning will be compared. It is expected that:

- PTF will obtain at least 50% of manual improvements, or even surpass them (>100%);
- PTF will require only a single or a few application runs, compared to effort timed in months for manual tuning.

## 7. AutoTune Project Work Plan




**Funding by European Union**
**FP7 Project no. 288038**
**Start date: October 15th, 2011**
**Duration: 36 months (2011-2014)**
**Total Cost: € 3.1 million**
**Contact: Michael Gerndt <gerndt@in.tum.de>**

Project Consortium: Technische Universität München, CAPS, Universitat Autònoma de Barcelona, NUI Galway OÉ Gaillimh, universität wien
Associate Partner: IBM
Funding: Science Foundation Ireland, e-infrastructure, Seventh Framework Programme