# cTuning.org: novel extensible methodology, framework and public repository to collaboratively address Exascale challenges

Grigori Fursin

# cTuning.org: novel extensible methodology, framework and public repository to collaboratively address Exascale challenges

Grigori Fursin
INRIA Saclay
France
grigori.fursin@inria.fr

Innovation in science and technology is vital for our society and requires faster, more power efficient and reliable computer systems. However, designing and optimizing such systems has become intolerably complex, ad-hoc, costly and error prone due to ever increasing number of available design and optimization choices combined with complex interactions between all software and hardware components, multiple strict requirements placed on characteristics of new computer systems, and a large number of ever-changing and often incompatible analysis and optimization tools.

Auto-tuning, run-time adaptation and machine learning based approaches have been demonstrating good promise to address above challenges for more than a decade but are still far from the widespread production use due to unbearably long exploration and training times, lack of a common experimental methodology, and lack of public repositories for unified data collection, analysis and mining.

At the same time educational and research methodology has hardly changed in decades: reproducibility and statistical meaningfulness of experimental results as well as sharing of data and tools is not even considered in most of the projects and numerous publications, in contrast with other sciences including physics, biology and AI. In fact, it is often impossible due to a lack of common and unified repositories, tools and data sets. Furthermore, there is a vicious circle since initiatives to develop common tools and repositories to unify, systematize, share knowledge (data sets, tools, benchmarks, statistics, models) and make it widely available to the research and teaching community are rarely funded or rewarded academically where a number of publications often matters more than the reproducibility and statistical quality of the research results.

Eventually, scientists, engineers and students are forced to resort to some intuitive, non-systematic, non-rigorous and error-prone experiments using ad-hoc tools, benchmarks and data sets. Furthermore, we witness slowed down innovation, a dramatic increase in development costs and time-to-market for the new HPC systems, enormous waste of expensive computing resources and energy, and diminishing attractiveness of computer engineering often seen as "hacking" rather than systematic science.

Therefore, I strongly advocate for a new long-term research and educational methodology including new publication model in computer engineering that favors sharing of data (applications, data sets, codelets), statistics about behavior of computer systems and models (classification, predictive modeling, run-time adaptation). Such collaborative approach allows the community to continuously validate, systematize and improve collective knowledge about computer systems and extrapolate it to build faster, more power efficient and reliable computer systems.

I present a long-term holistic and cooperative methodology and infrastructure for systematic characterization and optimization of computer systems through unified and scalable repositories of knowledge and crowdsourcing [1,2,3]. In this approach, multi-objective program and architecture tuning to balance performance, power consumption, compilation time, code size and any other important metric is transparently distributed among multiple users while utilizing any available mobile, cluster or cloud computer services.

Collected information about program and architecture properties and behavior is continuously processed online using statistical and predictive modeling techniques to build, keep and share only useful knowledge at multiple levels of granularity (online learning). Gradually increasing and systematized knowledge can be used to detect and characterize abnormal program or system behavior, or predict most profitable program optimizations, run-time adaptation scenarios and architecture configurations depending on user requirements.

Furthermore, the unified format of the collected data allows students and researchers from other interdisciplinary fields and without specialized knowledge in computers to validate existing or apply new statistical, data mining, classification and predictive modeling techniques to the collected "big data" and quickly suggest most profitable program optimizations, run-time adaptation scenarios or architecture designs that effectively balance performance, power consumption, return on investment and other characteristics depending on user's requirements. Collective methodology also enables a new publication model where all

experimental results and tools can be collaboratively and continuously validated and extended by the community.

Since 2007, Collective Tuning technology has been continuously validated and extended in several academic and industrial projects with IBM, ARC (Synopsys), CAPS Entreprise, Intel Exascale Lab and NCAR. I present a new version of the public, open-source infrastructure and repository (cTuning3 aka Collective Mind) to crowdsource characterization and optimization of computer systems using thousands of shared kernels, benchmarks and datasets from all previous projects combined with online learning plugins.

Collective Mind Framework includes 3 novel concepts:

1) Interactive interface to "open up" rigid tools and convert them into powerful interactive toolsets using light-weight event-based plugin framework (Interactive Compilation Interface and OpenME). ICI had been effectively used to convert GCC into self-tuning machine learning enabled compiler [3] and was added to the mainline GCC in 2010.

2) Technique for statistical program and architecture characterization through reactions to optimizations [3] or even semantically non equivalent code modifications similar to physics by removing or adding individual loops, instructions, code segments and threads to detect memory bottlenecks, contentions and other anomalies.

3) UNIDAPT framework to statically enable dynamic optimizations by combining a small set of pre-optimized versions of a code with online learning plugins to quickly select the most appropriate versions at run-time as a prediction or reaction to changing underlying (heterogeneous) architectures or varying program phases.

Collective Mind Infrastructure is in testing stage now and should be released in 2013. I hope that this collaborative methodology and infrastructure will help initiate systematization of the research, development and optimization methodology for the future Exascale systems while making computer engineering a "real" science.

## REFERENCES


[1] http://cTuning.org - public portal, infrastructure and repository for collaborative multi-objective online auto-tuning of computer systems

[2] Grigori Fursin. Collective Tuning Initiative: automating and accelerating development and optimization of computing systems. Proceedings of the GCC Summit'09, Montreal, Canada, June 2009

[3] Grigori Fursin and Olivier Temam. Collective Optimization: A Practical Collaborative Approach. ACM Transactions on Architecture and Code Optimization (TACO), December 2010, Volume 7, Number 4, pages 20-49

[4] Grigori Fursin, Yuriy Kashnikov, Abdul Wahid Memon, Zbigniew Chamski, Olivier Temam, Mircea Namolaru, Elad Yom-Tov, Bilha Mendelson, Ayal Zaks, Eric Courtois, Francois Bodin, Phil Barnard, Elton Ashton, Edwin Bonilla, John Thomson, Chris Williams, Michael O'Boyle. MILEPOST GCC: machine learning enabled self-tuning compiler. International Journal of Parallel Programming (IJPP), June 2011, Volume 39, Issue 3, pages 296-327