

## ABSTRACTS AUTO-DASP

### TRANSPARENT AUTONOMICITY FOR OPENMP APPLICATIONS

Daniele De Sensi and Marco Danelutto

One of the key needs of an autonomic computing system is the ability to monitor the application performance with minimal intrusiveness and performance overhead. Several solutions have been proposed, differing in terms of effort required by the application programmers to add autonomic capabilities to their applications. In this work we extend the Nornir autonomic framework, allowing it to transparently monitor OpenMP applications thanks to the novel OpenMP Tools (OMPT) API. By using this interface, we are able to transparently transfer performance monitoring information from the application to the Nornir framework. This does not require any manual intervention by the programmer, which can seamlessly control an already existing application, enforcing any performance and/or power consumption requirement. We evaluate our approach on some real applications from the PARSEC and NAS benchmarks, showing that our solution introduces a negligible performance overhead, while being able to correctly control applications' performance and power consumption.

### PARALLELIZATION OF MASSIVE MULTIWAY STREAM JOINS ON MANYCORE CPUS

Constantin Pohl and Kai-Uwe Sattler

Joining a high number of data streams efficiently in terms of required memory and CPU time still poses a challenge. While binary join trees are very common in database systems, they are mostly unusable for streaming queries with tight latency constraints when the number of streaming sources is increasing. Multiway stream joins, on the other hand, are very suitable for this task since they are mostly independent of the non-optimal ordering of join operators or huge intermediate join results.

In this paper, we discuss challenges but also opportunities for multiway stream joins for modern hardware, especially manycore processors. We describe different parallelization and optimization strategies to allow a streaming query to join up to 256 streams on a single CPU while keeping individual tuple response time and also memory footprint low. Our results show that a multiway join can perform magnitudes faster than a binary join tree. In addition, further tuning for efficient parallelism can improve performance again for a factor up to a magnitude.

### MINIMIZING SELF-ADAPTATION OVERHEAD IN PARALLEL STREAM PROCESSING FOR MULTI-CORES

Adriano Vogel, Dalvan Griebler and Luiz Gustavo Fernandes

Stream processing paradigm is present in several applications that apply computations over continuous data flowing in the form of streams (e.g., video feeds, image, and data analytics). Employing self-adaptivity to stream processing applications can provide higher-level programming abstractions and autonomic resource management. However, there are cases where the performance is suboptimal. In this paper, the goal is to optimize parallelism adaptations in terms of stability and accuracy, which can improve the performance of parallel stream processing applications. Therefore, we present a new optimized self-adaptive strategy that is experimentally evaluated. The proposed solution provided high-level programming abstractions, reduced the adaptation overhead, and achieved a competitive performance with the best static executions.

### A FULLY DECENTRALIZED AUTOSCALING ALGORITHM FOR STREAM PROCESSING APPLICATIONS

Mehdi Belkhiria Cédric Tedeschi

Stream Processing deals with the efficient, real-time processing of continuous streams of data. Stream Processing engines ease the development and deployment of such applications which are commonly pipelines of operators to be traversed by each data item. Due to the varying velocity of the streams, autoscaling is needed to dynamically adapt the number of instances of each operator. With the advent of geographically-dispersed computing platforms such as Fog platforms, operators are dispersed accordingly, and autoscaling needs to be decentralized as well. In this paper, we propose an algorithm allowing for scaling decisions to be taken and enforced in a fully-decentralized way. In particular, in spite of scaling actions being triggered concurrently, each operator maintains a view of its neighbours in the graph so as no data message is lost. The protocol is detailed and its correctness discussed. Its performance is captured through early simulation experiments.

### ADAPTIVE CROWN SCHEDULING FOR STREAMING TASKS ON MANY-CORE SYSTEMS WITH DISCRETE DVFS

Christoph Kessler, Sebastian Litzinger and Jörg Keller

Abstract. We consider temperature-aware, energy-efficient scheduling of streaming applications with parallelizable tasks and throughput requirement on multi-/many-core embedded devices with discrete dynamic voltage and frequency scaling (DVFS). Given the few available discrete frequency levels, we provide the task schedule in a conservative and a relaxed form so that using them adaptively decreases power consumption, i.e. lowers chip temperature, without hurting throughput in the long run. We support our proposal by a toolchain to compute the schedules and evaluate the power reduction with synthetic task sets.



Submission deadline for the Euro-Pas PhD Symposium has been extended to 27 May, 2023. Click here for more information - <https://t.co/wWxislCJSC>

17.05.2023 - 11:51

The Euro-Par PhD Symposium is a welcoming and supportive forum for PhD students to present their work. Click here for more information: <https://t.co/wWxislCJSC>

04.04.2023 - 09:25

Submit your paper for EURO-PAR 2023 Workshops and Minisymposia! Click here for more information. <https://t.co/UEseXWb3Dz>

07.03.2023 - 08:18

Abstract submission is due tomorrow 24 Feb, 2023 Click here for more information. <https://t.co/eH2C9CRZA3>

23.02.2023 - 08:18

CONTACT US

### HOSTS



### SPONSORS



SHARE ON:



