

The TURBO Diaries: Application-controlled Frequency Scaling Explained

Jons-Tobias Wamhoff
Stephan Diestelhorst
Christof Fetzer

*Technische Universität Dresden,
Germany*
first.last@tu-dresden.de

Patrick Marlier
Pascal Felber

*Université de Neuchâtel,
Switzerland*
first.last@unine.ch

Dave Dice

Oracle Labs, USA
first.last
@oracle.com

Abstract: Most multi-core architectures nowadays support dynamic voltage and frequency scaling (DVFS) to adapt their speed to the system’s load and save energy. Some recent architectures additionally allow cores to operate at boosted speeds exceeding the nominal base frequency but within their thermal design power.

We propose a general-purpose library that allows selective control of DVFS from user space to accelerate multi-threaded applications and expose the potential of heterogeneous frequencies. We analyze the performance and energy trade-offs using different DVFS configuration strategies on several benchmarks and real-world workloads. With the focus on performance, we compare the latency of traditional strategies that halt or busy-wait on contended locks and show the power implications of boosting of the lock owner. We propose new strategies that assign heterogeneous and possibly boosted frequencies while all cores remain fully operational. This allows us to leverage performance gains at the application level while all threads continuously execute at different speeds. Our in-depth analysis and experimental evaluation of current hardware provides insightful guidelines for the design of future hardware power management and its operating system interface.

Software development is confronted with the prediction that future processors will shift from designs with increasing core counts towards heterogeneous cores that accelerate special tasks. The reason is that substantial parts of the on chip resources must be powered off due to limitations in power delivery and thermal cooling. One of the key challenges will be efficient distribution of power among the cores of the chip. As a mid-term solution, the power distribution can be achieved by dynamic voltage and frequency scaling (DVFS) and power gating such that the cores run at asymmetric speeds.

Already today, multi-core architectures support DVFS. Typically, this feature is used to save energy during periods of low load or to boost peak loads. Thereby, the frequency and voltage is adjusted automatically by the operating system and the processor. Our goal towards efficiency is to enable applications to gain control over the hardware such that it can exploit its asymmetric properties to the underlying hardware. The efficiency is achieved by keeping all cores active and allowing a subset of the cores to operate at boosted speeds exceeding the nominal base frequency within the thermal design power of the processor.

We propose a library, named TURBO [WDF⁺ 14], that allows the programmatic control

of DVFS from user space to accelerate multi-threaded applications and expose the potential of heterogeneous frequencies. The library abstracts from the low-level control of the hardware and provides a convenient interface to configure the processor cores' speed according to the workload characteristics of an application.

To understand and reason about the impact of DVFS control, we first give an overview of the DVFS implementation of the hardware and then study the properties of adjusting the DVFS states. This includes the cost to trigger DVFS transitions and the latency until the processor finished the transition to a new frequency. We analyze the performance and energy trade-offs using different DVFS configuration strategies on several benchmarks and real-world workloads. With the focus on performance, we compare the latency of traditional strategies that halt or busy-wait on contended locks and show the power implications of boosting of the lock owner. We propose new strategies that assign heterogeneous and possibly boosted frequencies while all cores remain fully operational. This allows us to leverage performance gains at the application level while all threads continuously execute at different speeds. We also derive a model to help developers decide on the optimal DVFS configuration strategy, e.g., for lock implementations.

We apply the TURBO library in order to speed up real-world application and to validate our model. In particular, we show limitations of performance gains when the instructions per cycle depend on the core frequency and investigate to communicate to cores that execute at a different frequency. We also give insightful guidelines for the design of future hardware power management and its operating system interface.

All the details and results of our in-depth study can be found in the original publication [WDF⁺14].

References

- [WDF⁺14] Jons-Tobias Wamhoff, Stephan Diestelhorst, Christof Fetzer, Patrick Marlier, Pascal Felber, and Dave Dice. The TURBO Diaries: Application-controlled Frequency Scaling Explained. In *2014 USENIX Annual Technical Conference*, USENIX ATC'14, pages 193–204, Berkeley, CA, USA, June 2014. USENIX Association.