Heterogeneity Beyond Hybrid Architectures:
a Kernel for the Tomahawk

Nils Asmussen **Marcus Völp**
Technische Universität Dresden
Operating-Systems Group
<name>.<surname>@tu-dresden.de

Most of today's commercial-off-the-shelf multi-core systems already exhibit a limited form of heterogeneity by combining traditional general purpose cores with more or less integrated graphics processing units (GPUs). Today, this hybrid architecture is used both stationary in desktops or servers and in mobile devices. We believe future systems will confront us with much more heterogeneity. Because first, in their respective domains, special purpose accelerators show a performance and energy advantage over general purpose solutions of typically one or two orders of magnitude. Second, to compensate the effects of dark silicon, computer architects propose designs with a mixture of largely heterogeneous processing elements [2, 3]. And third, the disabling of faulty circuits to compensate reduced yields leads to increasing functional heterogeneity [1].

But how should the system architecture look like to support a wide variety of cores? How can specialized processing elements be integrated as first-class citizens, i.e. how can we run untrusted code on them in an isolated fashion? Today, each core has to provide the features an operating-system (OS) kernel requires to run on it. That is, the core needs to provide a privileged mode, exceptions to trap into the kernel, an MMU to isolate applications and instructions to manage the TLB. Features that do not necessarily exist on, e.g., accelerators or FPGAs. How can we relieve the hardware vendors from the requirement to provide these features and still integrate the cores as first-class citizens?

And finally, how does the integration of special purpose cores impact OS design? If not all cores provide support to run an OS, how can we still run untrusted code on them and how can the applications on these cores use OS functionality?

In this work-in-progress talk, we report on our findings and design decisions to support wildly heterogeneous processing elements. We sketch both the hardware extensions that allow us to uniformly control and isolate arbitrary types of cores and our microkernel called M3 (microkernel for minimalist many-cores or L4 +/- 1).

References

[1] S. Borkar. Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *Micro, IEEE*, 25(6):10–16, Nov 2005.

[2] Nikos Hardavellas, Michael Ferdman, Babak Falsafi, and Anastasia Ailamaki. Toward dark silicon in servers. *IEEE Micro*, 31(EPFL-ARTICLE-168285):6–15, 2011.

[3] Michael Taylor. A landscape of the new dark silicon design regime. 2014.