XBX for SUPERCOP-eBASH

e**X**ternal **B**enchmarking e**X**tension

Christian Wenzel-Benner / Jens Gräf

SUPERCOP, eBASH, eBACS

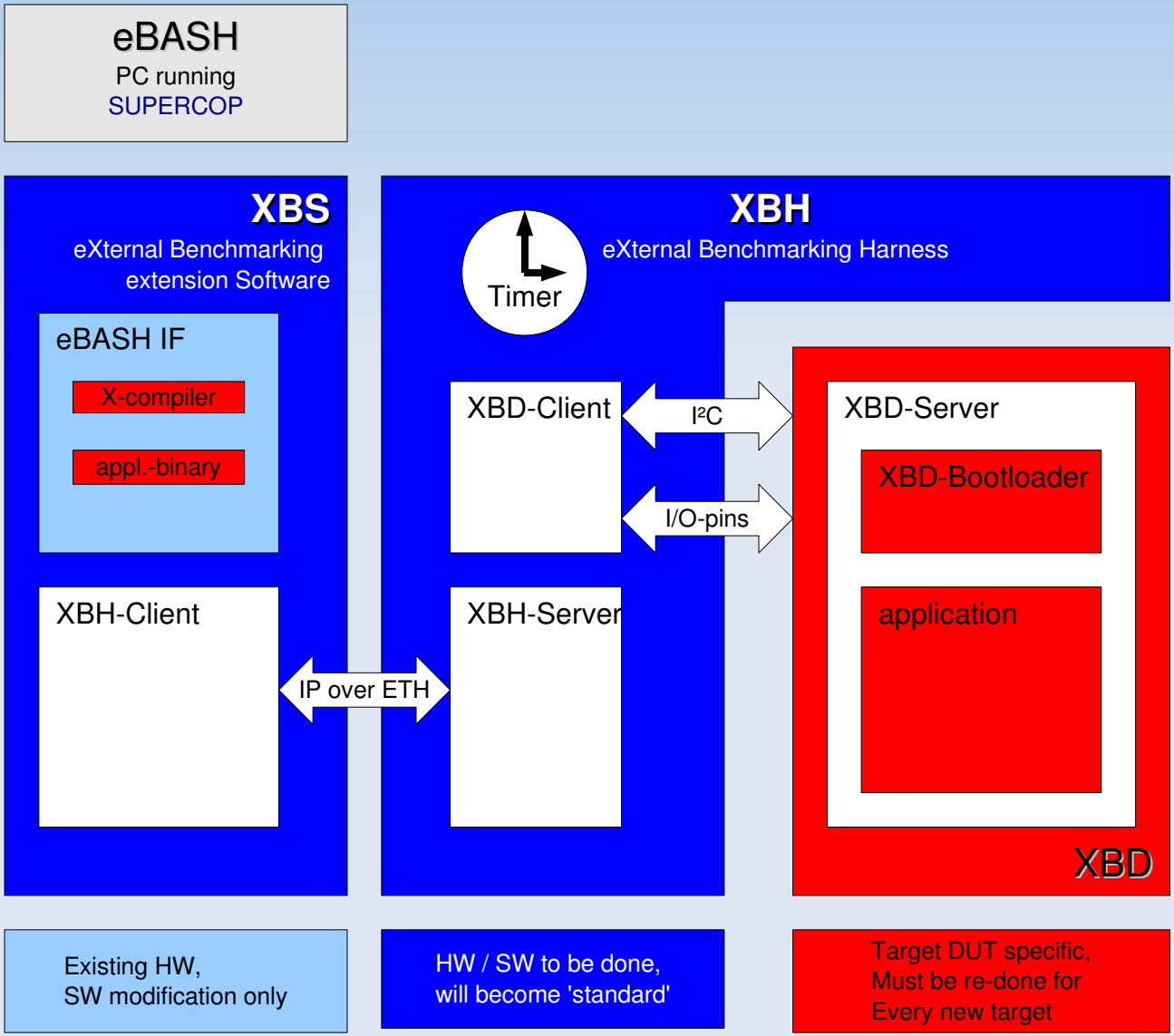
SUPERCOP is a toolkit developed by the VAMPIRE lab for measuring the performance of cryptographic software. SUPERCOP stands for **S**ystem for **U**nified **P**erformance **E**valuation **R**elated to **C**ryptographic **O**perations and **P**rimitives; the name was suggested by Paul Bakker.

Daniel J. Bernstein and Tanja Lange (editors). eBACS: ECRYPT Benchmarking of Cryptographic Systems. http://bench.cr.yp.to

Overview

- General idea from SUPERCOP: Compile/Try/Measure
- For each hash-algorithm from SUPERCOP
- Build binaries for each implementation-compiler pair
- Do quick benchmark for each binary
- Do a more detailed benchmark run for fastest
- Fastest passing the checksum test that is
- Implementation-compiler-options triple

Block Diagram



- Cross compile for target (XBD)
- Send to XBH via Ethernet
- XBH uploads to XBD by bootloader

Build Binaries (Compile)

- XBX contains a list of possible compilers and settings
- For each implementation found in SUPERCOP
- For each detected compiler:
- Build a binary by compiling the hash algorithm together with our framework

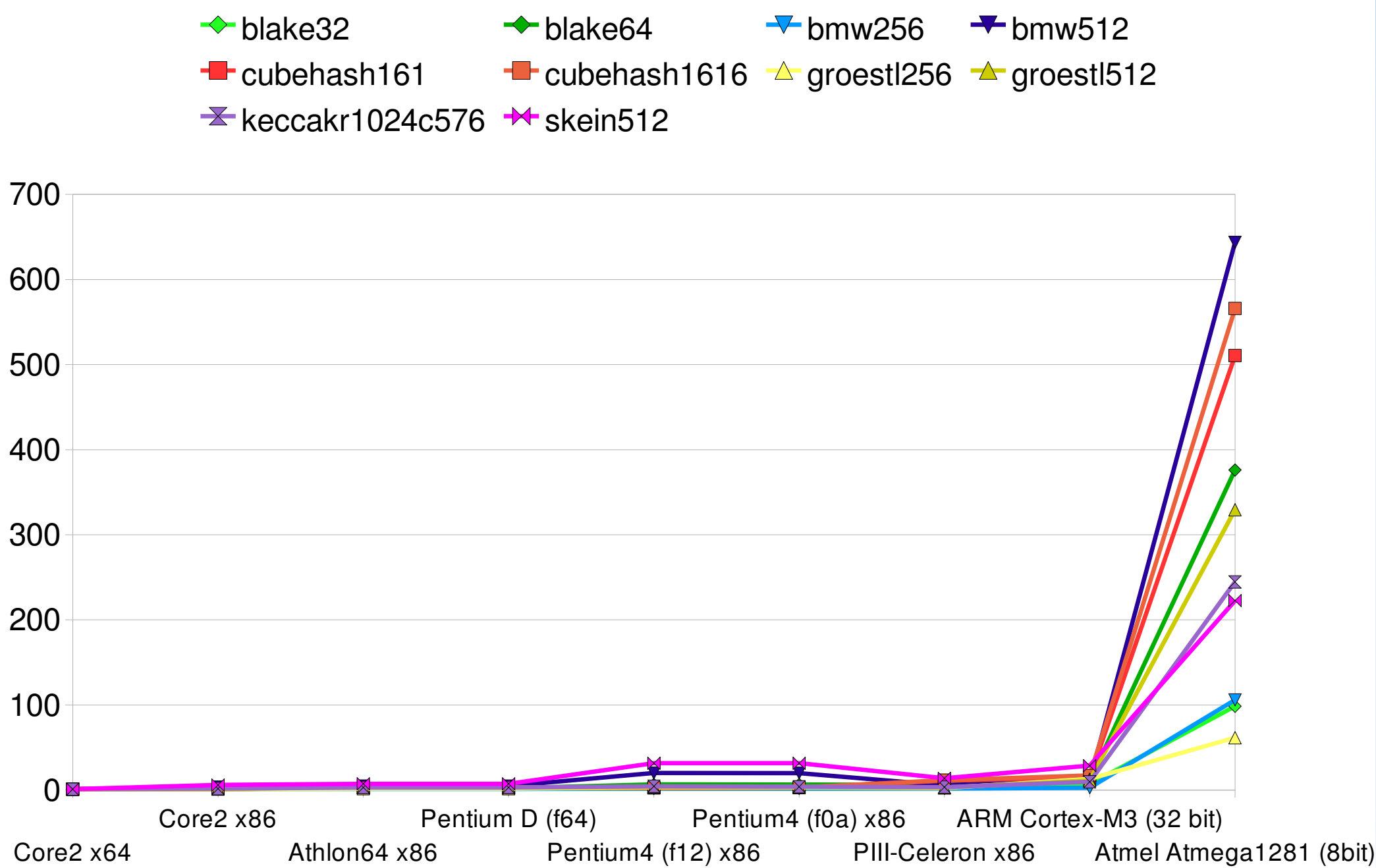
Quick Benchmark (Try)

- For each binary found built in the step above:
- Use our scripts and the XBH to download the code into the XBD
- Run known-answer tests to validate the generated code
- Record stack usage if available
- Measure the time it takes to hash 1536 bytes

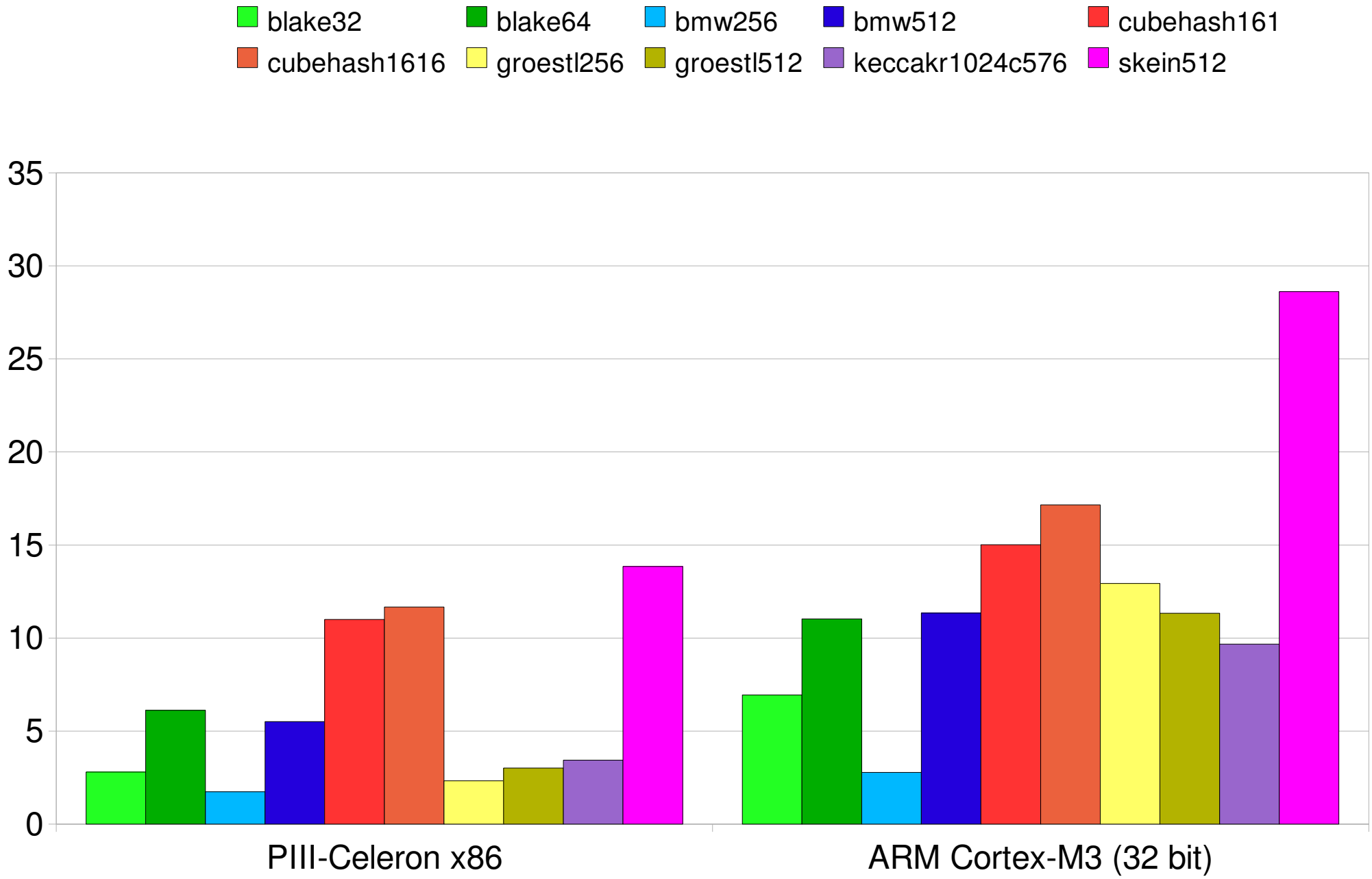
Detailed Benchmark (Measure)

- Take the fastest binary for a certain algorithm
- Measure the time it takes to hash different messages
- 1 byte, 32 bytes, 512 bytes, ...
- Output is (hopefully) in standard SUPERCOP format

Algorithm scaling from 64 to 8 bit



Intel Tualatin vs. ARM Cortex M3



Preliminary Conclusions

- XBX'd SHA3 candidates scale between 62X (Grøstl 256) and 644X (BMW 512) from Core2 64bit to Atmel AVR 8bit
- ARM Cortex M3 only has a 2X – 3X disadvantage on Pentium-III Tualatin (at ~60kGates vs. ~8MGates)
- Lots of optimisation potential for small devices