OOKAMI PROJECT APPLICATION

Date: **26/01/2021**

Project Title:

Software performance optimization through Parallelware tools for Fujitsu A64FX processor

Usage:

- x Testbed
- ☐ Production

Principal Investigator:

University/Company/Institute:	Appentra Solutions S.L.
Mailing address including country:	Centro de Investigación TIC. Campus de Elviña S/N. CP 15071. A Coruña. Spain
Phone number:	+34 677 788197
Email:	manuel.arenaz@appentra.com

Names & Email of initial project users:

Manuel Arenaz <manuel.arenaz@appentra.com> (IP)
Javier Novo <javier.novo@appentra.com>
Ivica Bogosavljevic <ivica.bogosavljevic@appentra.com>

Usage Description:

Appentra's Parallelware Analyzer is a cutting-edge static code analyzer specializing in parallelism and that is advancing the state of the art with respect to the commercial and open-source static code analyzers available today. It supports advanced source analysis and code rewriting capabilities specializing in SIM/vector and multi-threading for multicore CPUs and offloading for GPU accelerators using OpenMP and OpenACC. Work in progress since December 2020 aims at evaluating the performance of the performance optimization capabilities of Parallelware tools on high-performance processors from different vendors. The figures below showed the performance of the Canny edge detector image processing algorithm on Intel and AMD using OpenMP-enabled SIMD/vector and multi-threaded versions. The goal of this project is to work on the performance optimization of the Canny code for the Fujitsu A64FX processor using Arm SVE and OpenMP compiler directives. Other benchmarking codes can be also considered in order to better evaluate the benefits of extending Parallelware tools with Arm SVE support in the future.

Parallelware Analyzer (PWA) for SIMD: Intel AVX-512

The runtime was reduced up to 66% (~3x faster) on multicores using the Clang and GCC compilers

Use case: Canny (Image processing) Code versions using SIMD, multithreading and offloading parallelism	Clang 10.0 (seconds)	GCC 8.2 (seconds)
Canny SERIAL (serial version, maximum optimization without auto-vectorization)	11,08	12,03
Canny AUTO (auto-vectorized serial version, maximum optimization)	10,64 [3.9%] [1.04x]	11,63 [3.4%] [1.04x]
Canny PWA SIMD (*) (PWA SIMD + auto-vectorized serial version, maximum optimization)	**4,97 [55.1%] [2.23x]**	**5,61 [53.4%] [2.14x]**
Canny PWA MULTI+SIMD (PWA Multithreading+SIMD + auto-vectorized serial version, maximum optimization)	**3,72 [66.4%] [2.98x]**	**4,08 [66.1%] [2.95x]**
Canny PWA GPU+SIMD (PWA GPU+SIMD + auto-vectorized serial version, maximum optimization)	-	-

Higher % is faster

CPU: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz (EPEEC cluster)
Problem size: Image size 15360 x 8640.

(*) Performance optimization achieved by a senior performance software engineer and that is the expected maximum performance improvement to be provided by Parallelware Analyzer (PWA) SIMD.

appentra

Computational Resources:

Total node hours per year: Low (difficult to estimate)
Size (nodes) and duration (hours) for a typical batch job: 1 node, <1 hour
Disk space (home, project, scratch): Order of MBs/GBs

Personnel Resources:
No additional personnel resources required, apart from support in the usage of the Arm software stack available in Ookami.

Required software:
C/C++/Fortran development tools for Arm platforms including Arm's optimizing compiler, Clang and OpenMP. It would be desirable to have access to the latest versions of the tools.

If your research is supported by US federal agencies:

Agency:
Grant number(s):

Production projects:

Production projects should provide an additional 1-2 pages of documentation about how
(a) the code has been tuned to perform well on A64FX (ideally including benchmark data comparing performance with other architectures such as x86 or GPUs)
(b) it can make effective use of the key A64FX architectural features (notably SVE, the high-bandwidth memory, and NUMA characteristics)
(c) it can accomplish the scientific objectives within the available 32 Gbyte memory per node