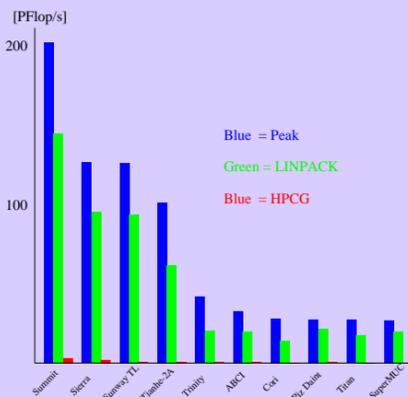


LOWAIN Project

LOW Arithmetic INTensity specific architectures

Running HPCG is computationally inefficient



| Computer | HPCG [PFlop/s] | Peak [PFlop/s] | LINPACK [PFlop/s] | Eff. [%] |
|-----------|----------------|----------------|-------------------|----------|
| Summit | 200.8 | 143.5 | 2.93 | 1.5 |
| Sierra | 125.7 | 94.6 | 1.80 | 1.4 |
| Sunway TL | 125.4 | 93.0 | 0.48 | 0.4 |
| Tianhe-2A | 100.7 | 61.4 | 0.58 | 0.6 |
| Trinity | 41.5 | 20.2 | 0.55 | 1.3 |
| ABCI | 32.6 | 19.9 | 0.51 | 1.6 |
| Cori | 27.9 | 14.0 | 0.36 | 1.3 |
| Piz Daint | 27.2 | 21.2 | 0.50 | 1.8 |
| Titan | 27.1 | 17.6 | 0.32 | 1.2 |
| SuperMUC | 26.9 | 19.5 | 0.21 | 0.8 |

Peak, LINPACK, and HPCG performance of the Top10 supercomputers (November 2018) - a graph and a table

LOWAIN assumptions and goals

LOWAIN assumptions:

- a simulation specific architecture is economically justified
- most simulation programs behave in a way similar to HPCG

LOWAIN goal: "Exascale-equivalent" computer

Summit-like exascale

Perform. estim. [PFlop/s]:
 DP peak 1000
 SP peak 2000
 DP HPCG ~15
 Simulations (DP) ~15-30
 Simulations (SP) ~50-60

"Exascale-equivalent"

Perform. estim. [PFlop/s]:
 DP peak 30-50
 SP peak 60-100
 DP HPCG ~15
 Simulations (DP) ~15-30
 Simulations (SP) ~50-60

F/B of Matrix-Vector Product

$$A_0 = M_{00} * a_0 + M_{01} * a_1 + M_{02} * a_2 + M_{03} * a_3$$

$$A_1 = M_{10} * a_0 + M_{11} * a_1 + M_{12} * a_2 + M_{13} * a_3$$

$$A_2 = M_{20} * a_0 + M_{21} * a_1 + M_{22} * a_2 + M_{23} * a_3$$

$$A_3 = M_{30} * a_0 + M_{31} * a_1 + M_{32} * a_2 + M_{33} * a_3$$

Each matrix element used only once (all accesses result in cache misses)
 Only two operations (MPY and ADD) done with any **non-zero** matrix element. (vector loads not considered)

Flop/Byte of Matrix-Vector Product
 2 operations/8 byte number < 0.25

DP HPCG Flop/Byte ratio is similar

Poor HPCG behavior is caused by low Flop/Byte ratio

| Processor | Memory Bandwidth [GB/s] | Enough Data for DP HPCG [GFlop/s] | Peak Performance [GFlop/s] | Bound to Efficiency [%] |
|---------------------------|-------------------------|-----------------------------------|----------------------------|-------------------------|
| NVIDIA Volta-100 | 900 | 0.25*900=225 | 7800 | 2.88 |
| Volta-100/NVLink | 300 | 0.25*300= 75 | 7800 | 0.96 |
| Intel Xeon Phi "KNL" | 480+120 | 0.25*600=150 | 3000 | 5.00 |
| KNL (using external DRAM) | 120 | 0.25*120= 30 | 3000 | 1.00 |

The processor-memory bandwidth performance limit and the peak performance

The first LOWAIN phase

The processor peak performance can not be fully used
 The LOWAIN program suggests **reducing the computing power and/or the number of cores of processors.**
 The first LOWAIN research goal is to determine how much by measuring Flop/Byte ratio of simulation programs.

Exploiting Flop/Byte ratio

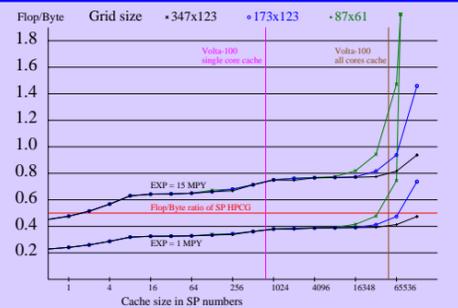
| Computer | Processor | Theoret. efficiency bound [%] | Measured efficiency [%] | % of use of the memory bandwidth bound [%] |
|----------|----------------------|-------------------------------|-------------------------|--|
| SX-ACE | Fujitsu SX-ACE | 25 | 11 | 44 |
| K | Fujitsu SPARC VIIIfx | 12 | 6 | 50 |
| Cori | Intel Xeon Phi "KNL" | 5.0 | 1.5 | 30 |
| Summit | NVIDIA Volta-100 | 2.9 | 1.5 | 52 |

The percentage of the use of the memory bandwidth when running the HPCG

The second LOWAIN phase

The real processor simulation performance is substantially worse than the memory bandwidth upper bound.
 The LOWAIN project suggests **using an intelligent memory controller** to make full use of the memory bandwidth upper bound.

Weather Research & Forecast



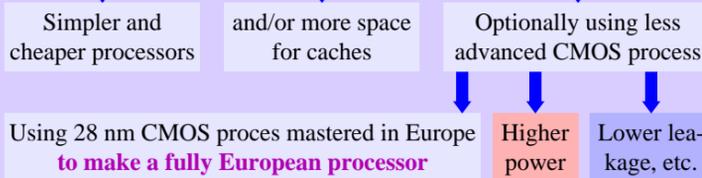
Flop/Byte of Microphysics Driver of Weather Research & Forecast as a function of the cache size (Single Precision configuration)
 A LOWAIN 1st phase result; input "Central Europe, June 6, 2013"
 Presented at General Assembly of European Geosci. Union, April 2019

Features of the Exascale-Equivalent Architecture

Very wide and fast memory bus to guarantee very high memory bandwidth
 It is not the goal of LOWAIN to prepare a HW design of a high-bandwidth memory bus, but to suggest measures to use a given bus optimally.

Reduced Number and/or Power of Processor Cores

Just as many cores as the memory bandwidth would keep busy



Using 28 nm CMOS proces mastered in Europe **to make a fully European processor**

Intelligent memory controller

Necessary to use efficiently the limited memory bandwidth. The standard pre-fetching and cache-miss procedures are too weak to take full use of simulation specific features

Main Program
 DO I=1,X
 A(2*I)=B(I)
 C(I+1)=D(I)+2
 ENDDO

Backbone
 <-B(1)
 A(2)->
 <-D(1)
 C(2)->
 <-B(2)
 A(3)->
 <-D(2)
 ...

Off-processor controller running the load/store backbone of the main program to deliver a data stream to/from the processor optimally and just-in-time.
 Very limited communication with the program cores.
 The present LOWAIN research shows that, in simulation programs, the backbone can run well ahead of the main program most of the time, and hence it has enough time to prepare the data flow for the processor

Pursued Approach and Methodology

1st Phase

Using standard profiling tools to measure execution times, the number of executed operations and the number of loads/stores across the processor-memory interface can be measured to determine the flop/byte ratio of studied programs. However, the number of loads/stores across the processor-memory interface depends on the cache sizes that are fixed when profiling at a given computer. Therefore, an emulator of a plain or optimized code with variable cache size is being developed for exact measuring of the flop/byte ratio dependence on the cache size

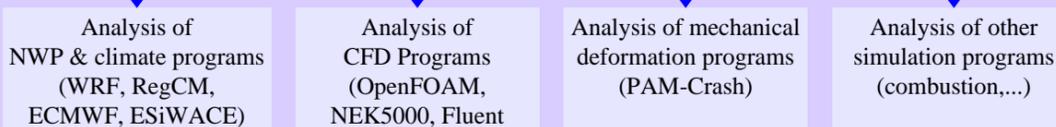
2nd Phase

- Study of patterns of processor-memory data traffic that are specific for computer simulations listed above and use them to design memory handling algorithms.
- Extend the emulator, developed in the first phase, to study the behavior and properties of different intelligent memory computers implementing the algorithms of the previous paragraph.
- Insert a low level model of the RISC-V architecture to the emulator to verify details of the LOWAIN processor design

The LOWAIN Project Roadmap

1st Phase (Jan 2019 - June 2020)

An emulator with variable cache size (June 2019)



2nd Phase (Jan 2020 - June 2021)

study of patterns of processor-memory data traffic (June 2020)

The 1st phase emulator extended to a model with an abstract memory controller (Oct 2020)

Evaluation of variants of smart memory controllers (Feb 2021)

Extension of the model by a low level model of RISC-V cores for final verification of the computer

The LOWAIN Project Consortium

Czech Technical University, Faculty of Information Technologies (Coordinator)
 Czech Technical University, Faculty of Mechanical Engineering
 Charles University, Department of Atmospheric Physics
 Charles University, Department of Applied Mathematics
 Skoda Auto, a.s.
 Mecas ESI (suggested)
 Codasip, s.r.o. (2nd phase)

Poster Author

Ludek Kucera
 LOWAIN Project
 Czech Technical University & Charles University
 Prague, Czech Republic
 ludek@kam.mff.cuni.cz