CISL Computational & Information Systems Laboratory

Evaluating the performance of the NEC Vector Engine and Cavium Thunder X2 on NCAR kernels

John Dennis, Brian Dobbins, Youngsung Kim

Application Scalability and Performance Group, CISL

NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

NSF

Motivation

- Impact of compiler choice on Cavium Thunder X2 performance
- Identify areas of potential collaboration to strengthen ARM software ecosystem
- Evaluate the durability of Intel Xeon and Xeon Phi optimizations
- Explore use of the NEC Vector Engine

Acknowledgements

- NEC
  - Masashi Ikuta and team
  - Tim Miller
  - Randy Pidhayny
- Cavium/ARM
  - Larry Wikelius and team
  - Ashok Bhat and Team
  - Srinath Vadlamani

Approach

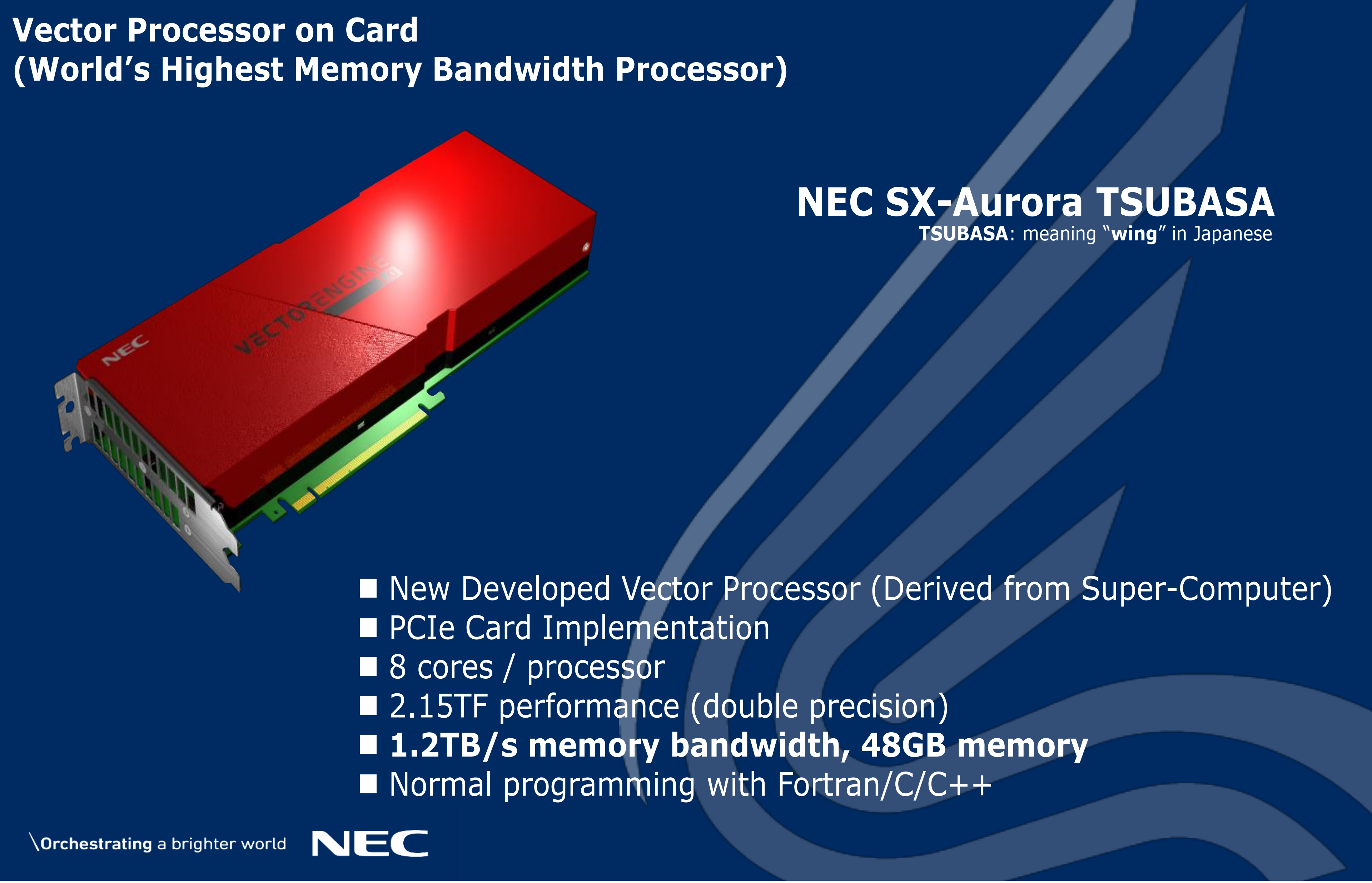
- STREAM2
- Elemental function (exp,log,pwr,sqrt)
- NCAR Kernels

Computational & Information Systems Laboratory

Platforms

- 2 x Intel Xeon E5-2697v4 Broadwell
  - 2.3 Ghz
  - 45 MB L3 cache
  - 36 cores
  - 64 GB, 2400 Mhz DDR4
- NEC VE 10-B
  - 1.4 Ghz
  - 16 MB L3 cache
  - 8 cores
  - 48 GB, HBM2
- 2 x Cavium Thunder X2
  - 2.2 Ghz
  - 32 MB L3 cache
  - 64 cores
- 2 x Intel Xeon Skylake Gold 6148
  - 2.4 Ghz
  - 27.5 MB L3 cache
  - 40 cores

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH



Vector Processor on Card
(World’s Highest Memory Bandwidth Processor)

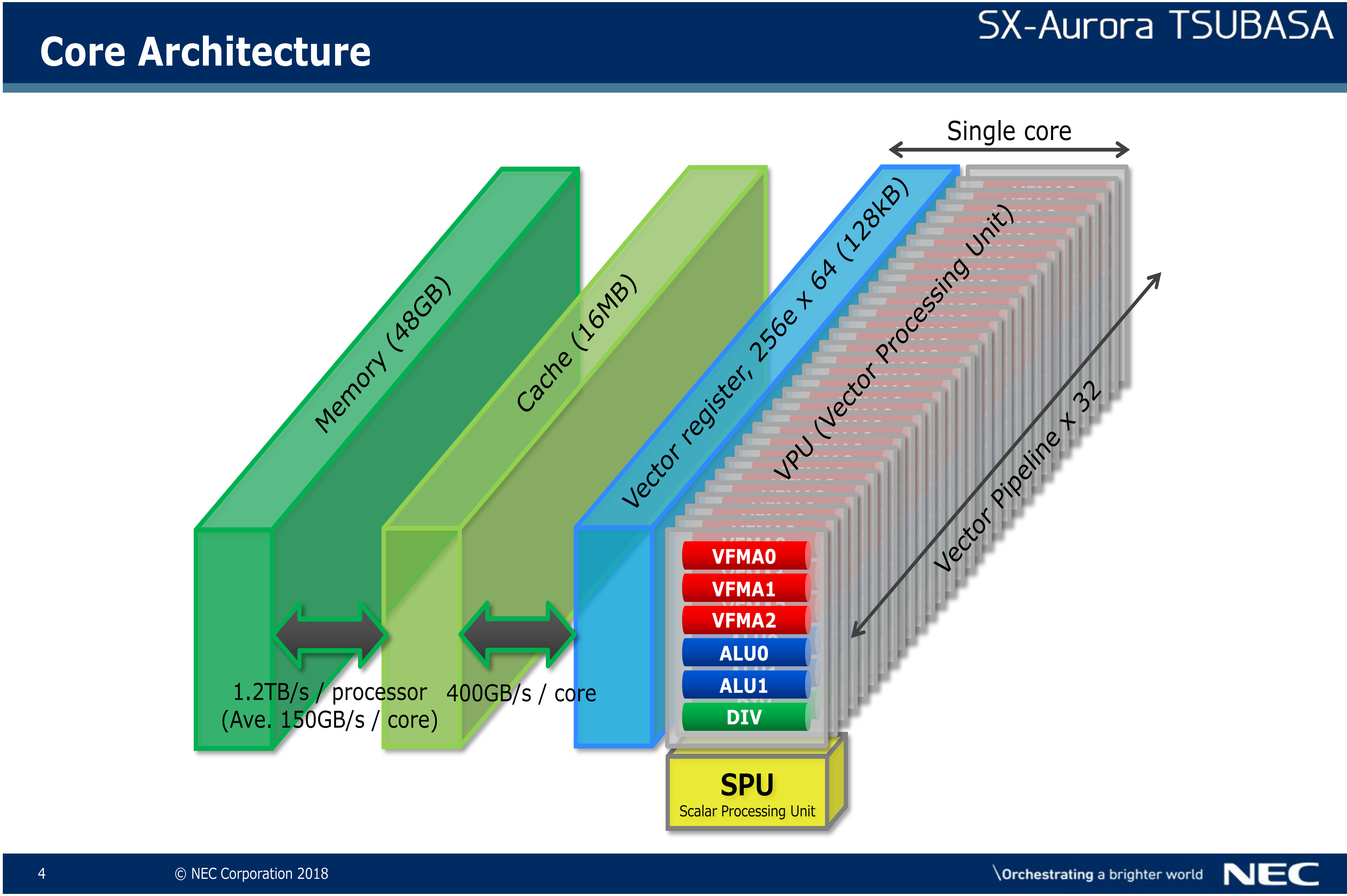
NEC SX-Aurora TSUBASA

TSUBASA: meaning “**wing**” in Japanese

- New Developed Vector Processor (Derived from Super-Computer)
- PCIe Card Implementation
- 8 cores / processor
- 2.15TF performance (double precision)
- **1.2TB/s memory bandwidth, 48GB memory**
- Normal programming with Fortran/C/C++

\Orchestrating a brighter world NEC

Image from NEC_SX_Aurora_TSUBASA\(WING\)_QuickStartGuide_20180713.pdf



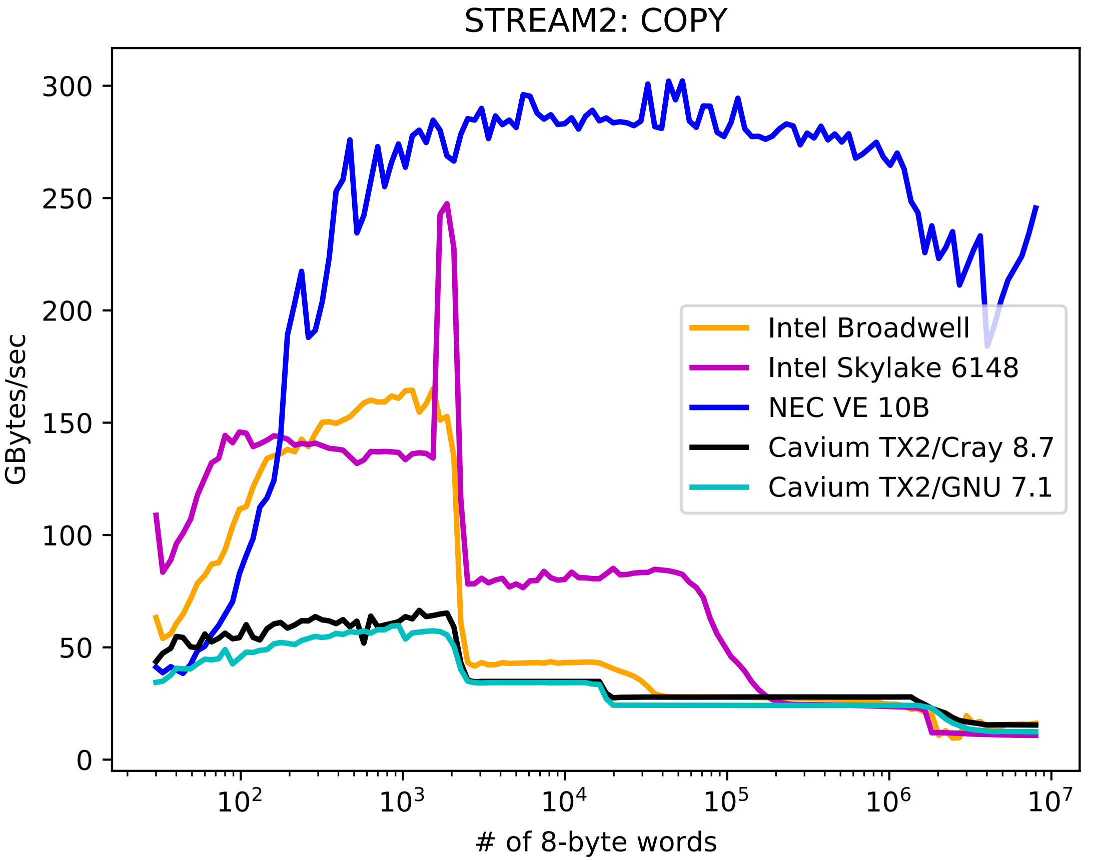
SX-Aurora TSUBASA

Core Architecture

4 © NEC Corporation 2018 \Orchestrating a brighter world NEC

Image from NEC_SX_Aurora_TSUBASA\(WING\)_QuickStartGuide_20180713.pdf

STREAM2: COPY (single core)



CISL Computational & Information Systems Laboratory

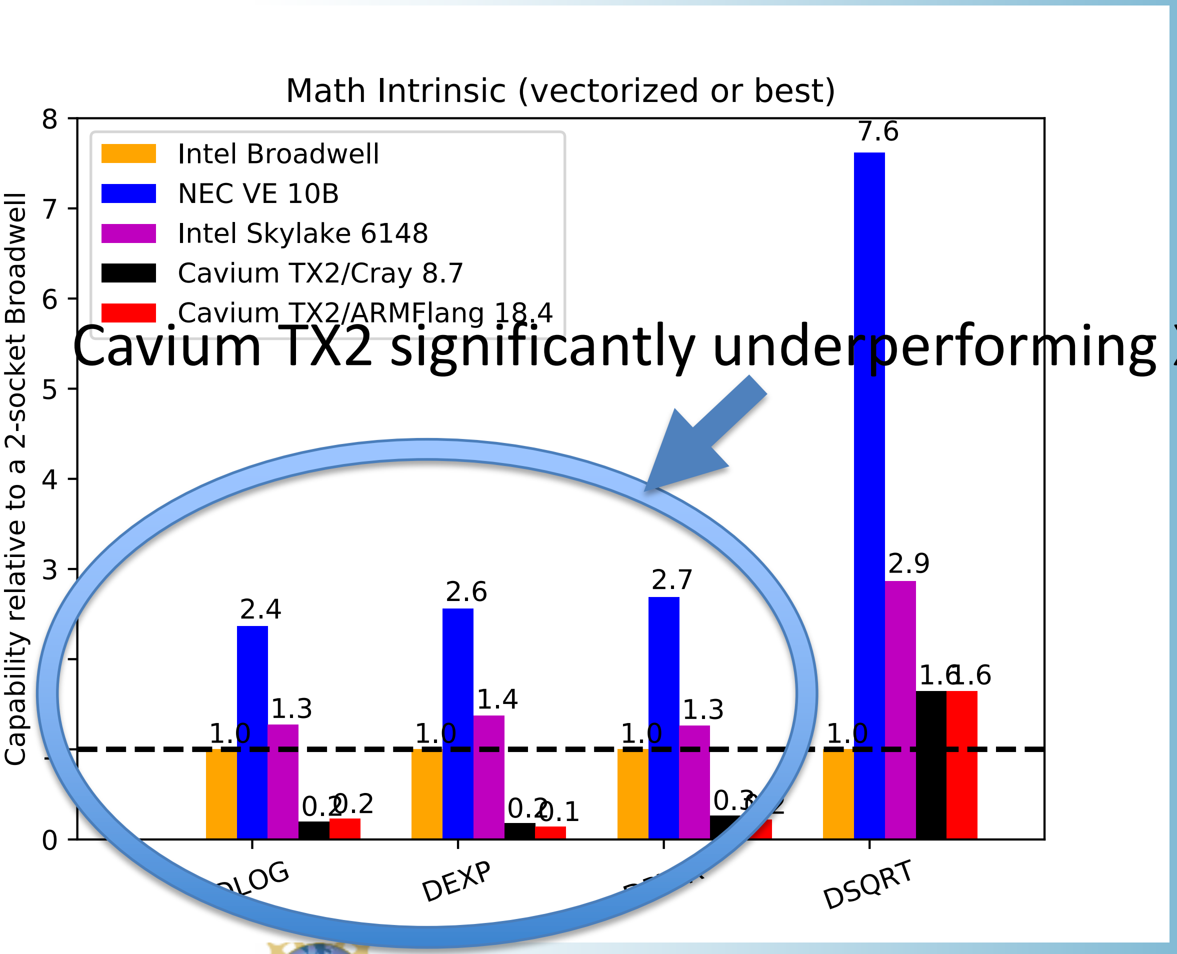
NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

NSF

Elemental function performance (elefunt)

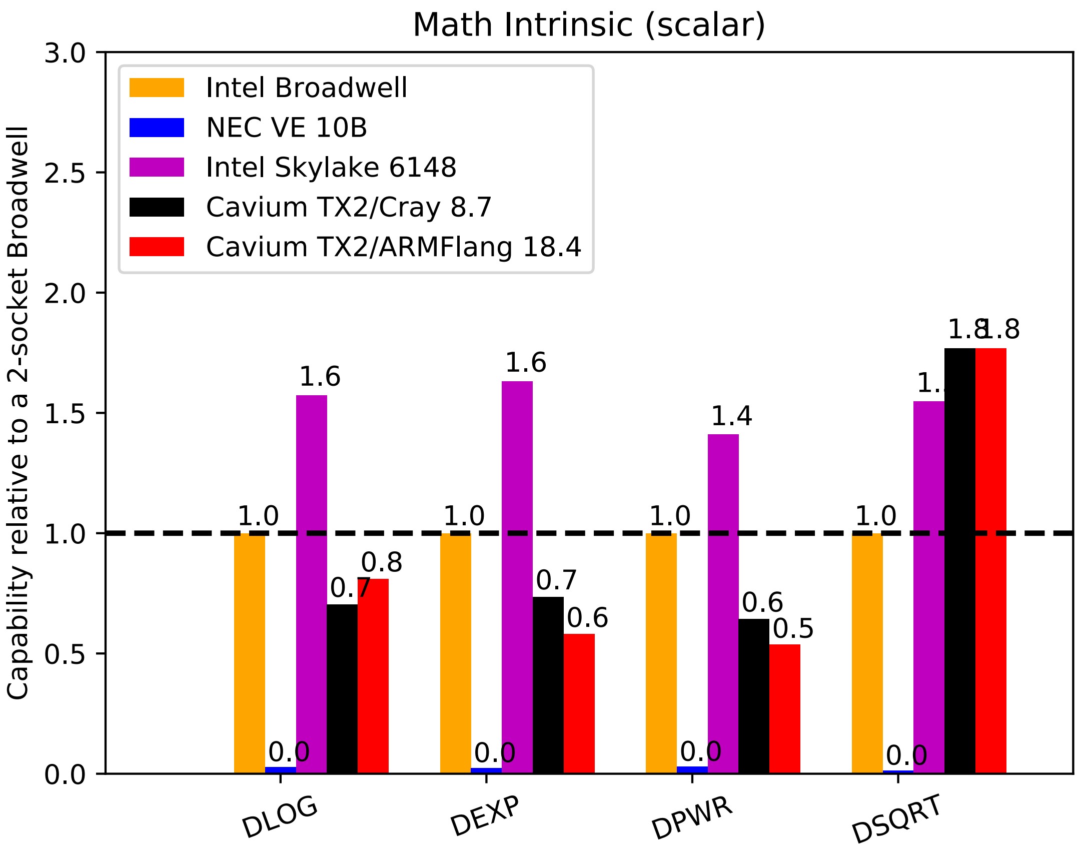
- CESM physics code utilizes large number of elemental function calls
- Poorly performing math intrinsics can negatively impact overall CESM
  - Simple way for vendors to improve performance

Elemental math function rate



Cavium TX2 significantly underperforming Xeon

Elemental math function rate (scalar)



Vectorization on VE is critical!

Description of NCAR Kernels

- Selection of kernels extracted from NCAR applications (https://github.com/NCAR/kernelOptimization)
  - Created using KGEN (https://github.com/NCAR/KGen)
  - Expensive sections of computational code
  - Representative input data
  - Built in verification testing

Description of NCAR Kernels

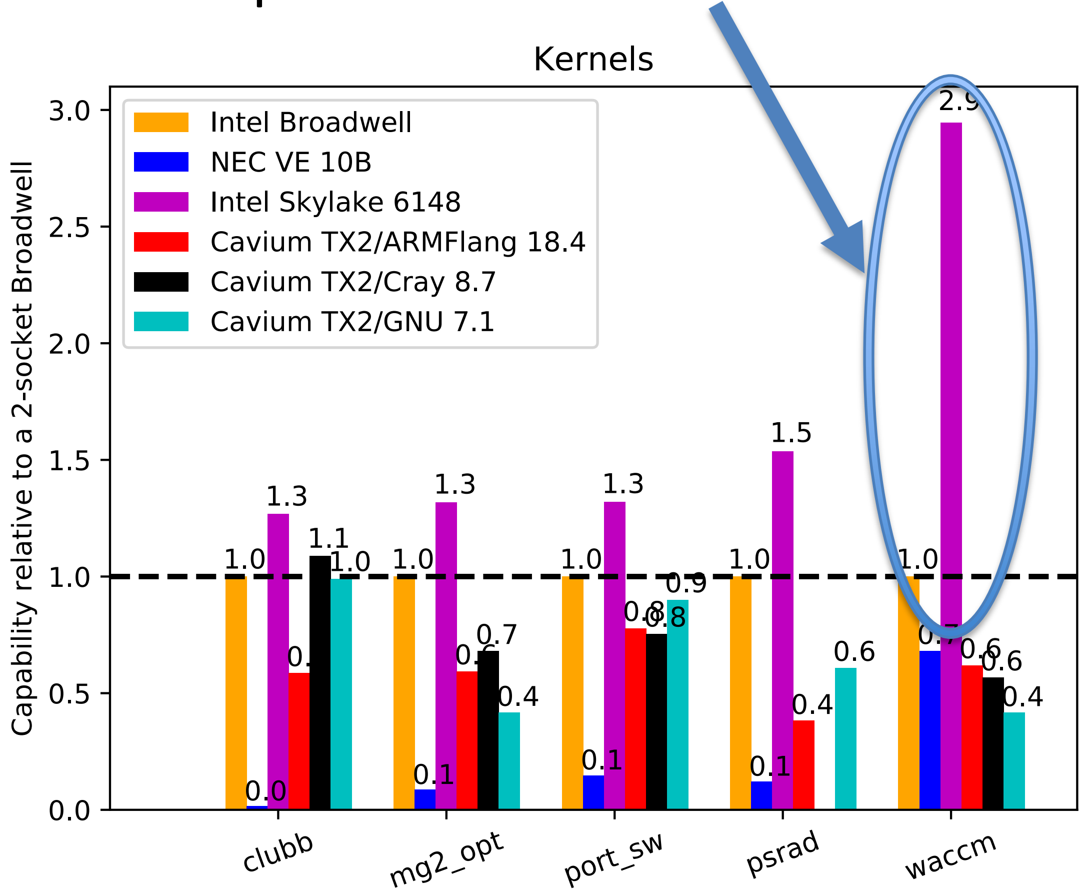
- CLUBB:
  - Cloud macrophysics and turbulence
  - Status: straight from scientists
  - Single column abstraction ☹
- MG2_opt:
  - Version 2 of the Morrison Gettleman microphysics
  - Status: Extensive optimization for Intel Xeon & Xeon Phi [MC5]
  - Math intrinsic performance sensitive
- PORT_SW:
  - Short-wavelength radiation from RRTMG
  - Status: Original pre-optimized code
- PSRAD:
  - Long-wavelength radiation from RRTMG
  - Status: Optimization for Intel Xeon & Xeon Phi [MC5]

Description of NCAR Kernels

- WACCM:
  - Chemistry solver from the Whole Atmosphere Climate Community Model
  - Status: Extensive optimizations for Intel Xeon & Xeon Phi
  - **Adjustable inner-loop dimension**
- CESM2_MG2
  - Version 2 of the Morrison Gettleman microphysics
  - Status: **Extensive optimization for NEC Vector Engine** [MC8]
  - Math intrinsic performance sensitive

NCAR Kernels (normalized to 2-socket Broadwell)

Combined impact of AVX512 and L2 cache



Computational & Information Systems Laboratory

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

WACCM

- Auto-generated linear solver from chemical specification
- Resurrected version of auto-generator from 1990’s to support the creation of vector version
- 99.8% vectorized (NEC VE)
- Very demanding of L2 cache

Can MG2 be refactored to be long vector ready?

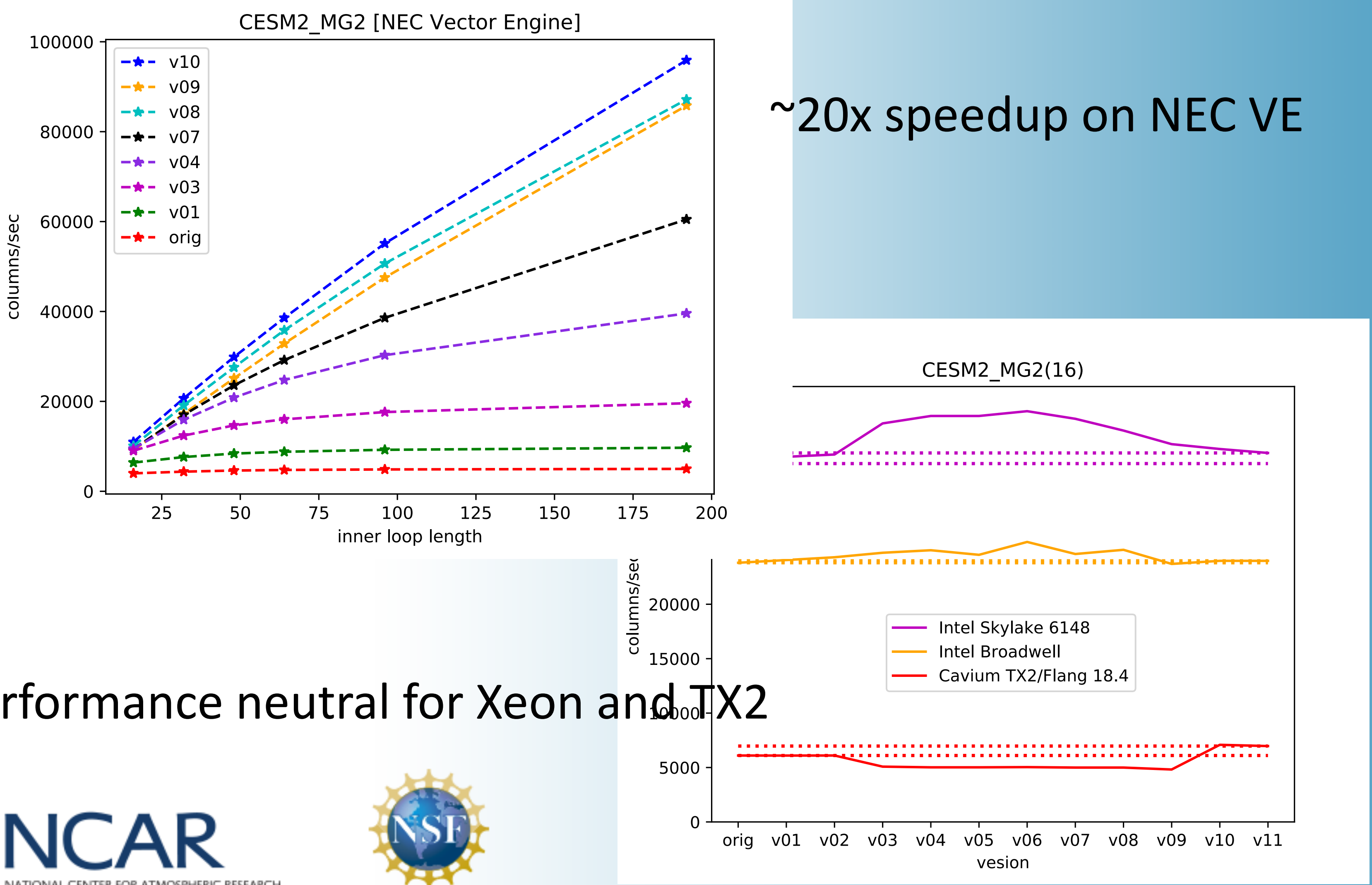
Refactor the MG2 kernel

- MG2_opt:
  - Extensively optimized for Xeon & Xeon Phi 2.3x speedup
  - Minimize code changes
- CESM2_MG2:
  - Part of CESM2 release code
  - Initial version: NEC VE achieves 3.7% of Broadwell node ☹
  - Address several key questions
    - How difficult to refactor to support long vector length for NEC VE?
    - Impact on Intel Xeon, Cavium Thunder X2 performance?

Refactoring CESM2_MG2

- Used vendor supplied GAMMA function
  - Single use of __NEC__ cpp ifdef
- Vectorize nearly all subroutines
- Calculate everywhere and mask out results
  - Did not check for vector occupancy
- Inline functions
- Input datasets with various lengths of inner loops (16,32,48,64,96, and 192)
- Currently 96% vectorized

Performance evolution of CESM2_MG2 kernel



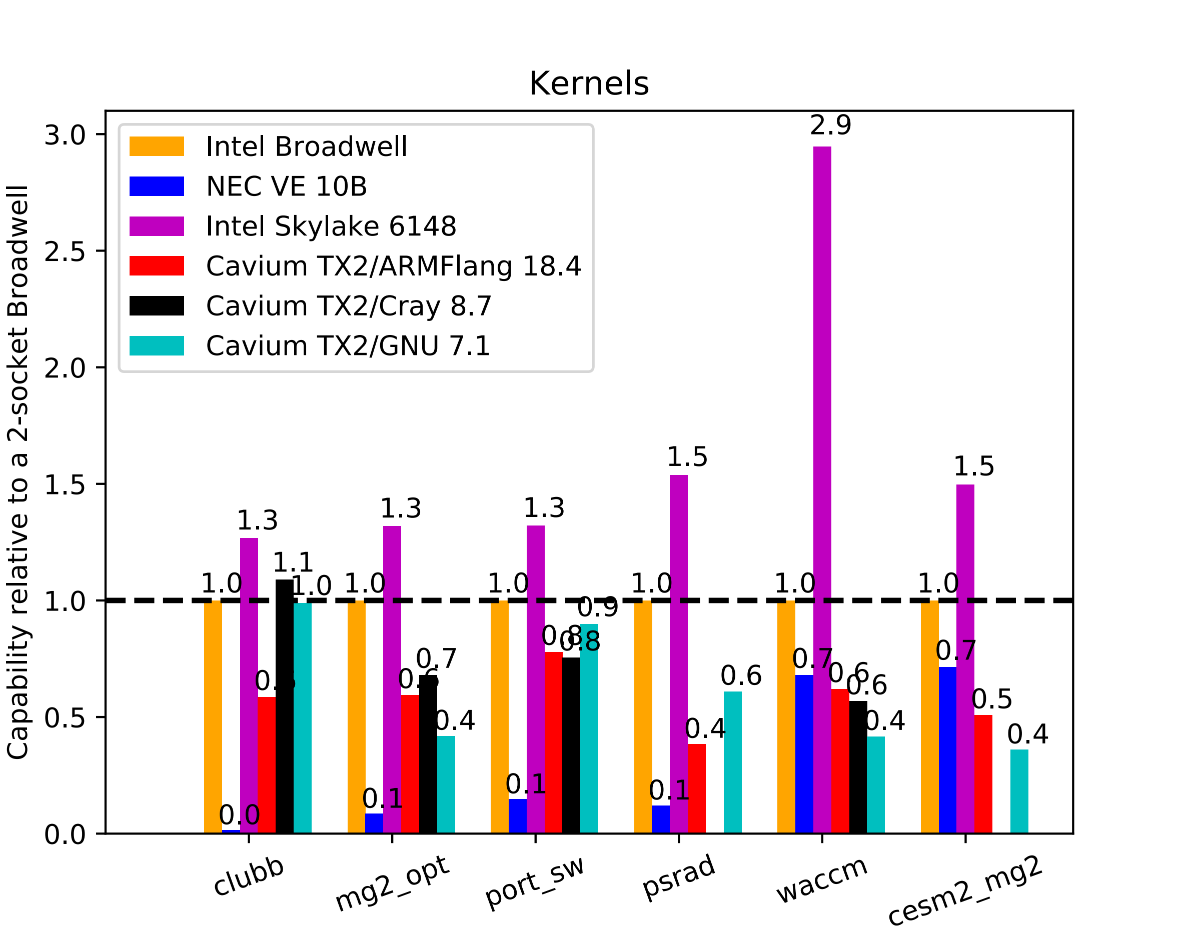
~20x speedup on NEC VE

Performance neutral for Xeon and TX2

Code optimization on the NEC VE

- Ftrace capability
  - Good
    - Augmented gprof capability
    - Very easy to use
    - Provides % vectorization, vector length, LLC efficiency, etc
  - Bad
    - Cost of built-in Fortran get lost (TRIM, scalar LOG)
- Compiler reports
  - Simple, easy to use
  - Look for “Unvectorized loop”
  - Good concordance between compiler report, ftrace report and runtime

NCAR Kernels (normalized to 2-socket Broadwell)



Conclusions

- Cavium Thunder X2
  - 40-110% of capability of Broadwell node on multiple kernels
  - Improvement to math intrinsics should help significantly (October 2018 compiler release)
- NEC Vector Engine
  - 7-70% of capability of Broadwell node on multiple kernels
  - Sufficiently long vector length is critical to performance
  - Only a single kernel was long vector length ready
  - Robust easy to use performance optimization environment
- Long vector length ready code basis of reduced precision versions

Questions?

dennis@ucar.edu