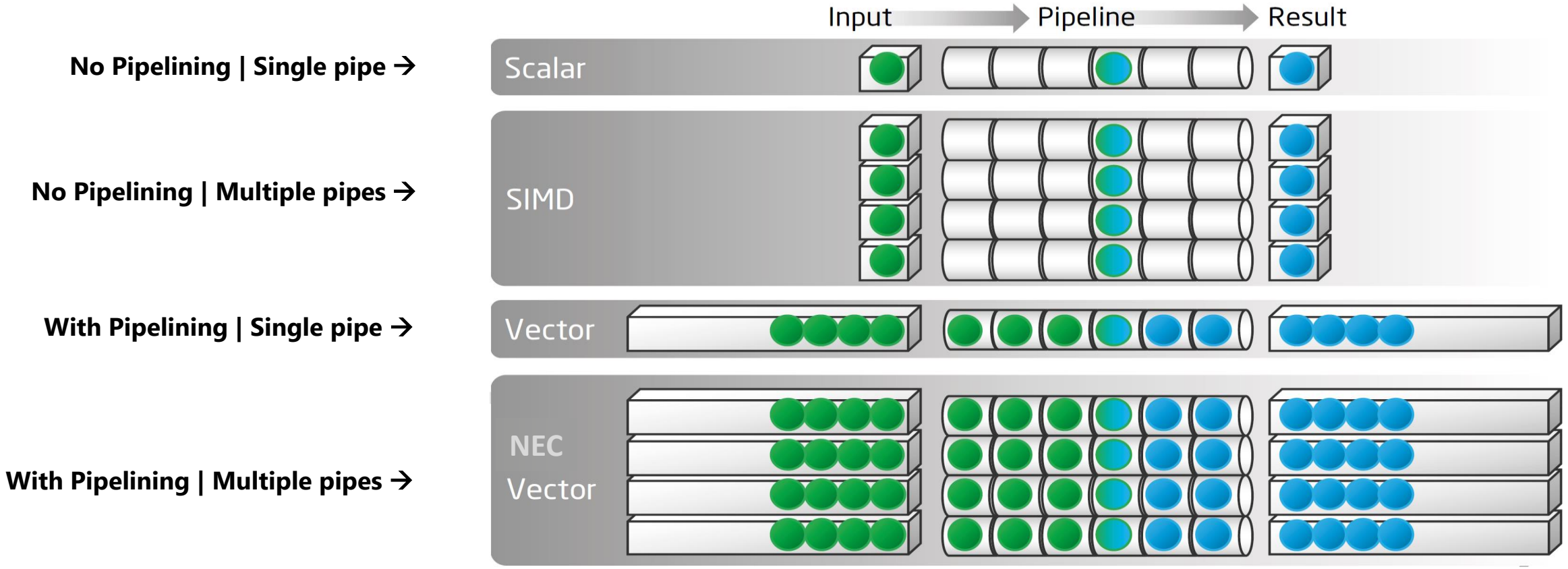
Accelerating complex risk analysis/STAC-A2™ workloads with easy programming using NEC's vector technology

Raghunandan Mathur
NEC Corporation India

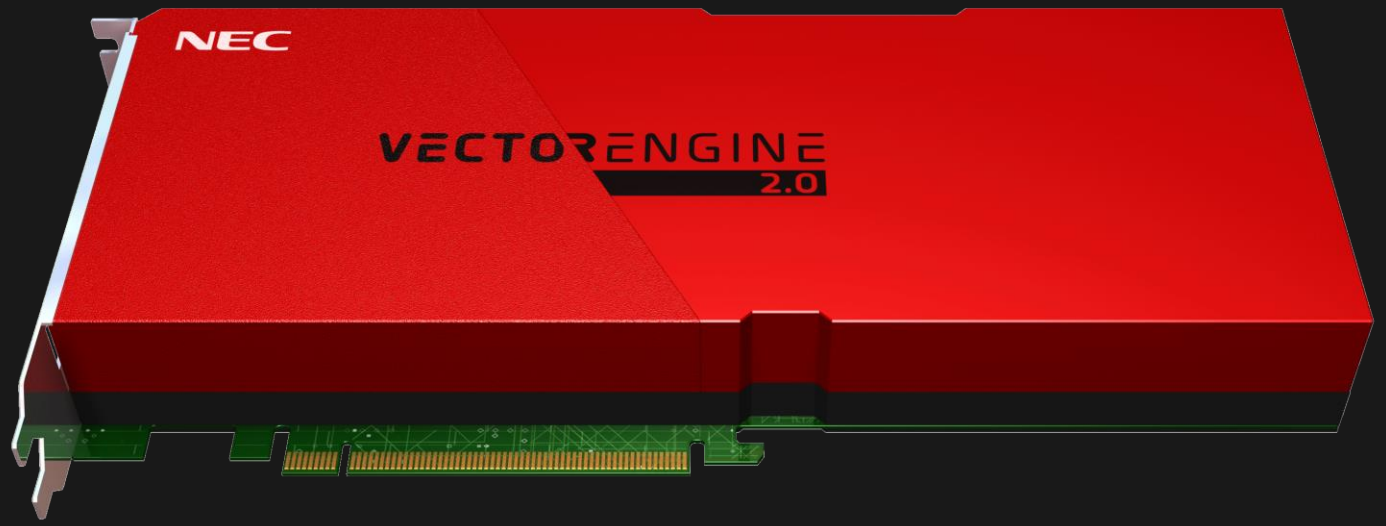
Global STAC Live
18-May-2020

The vector processing philosophy

- A processor architecture that processes a 'Vector' as a single data element.
- Arbitrarily large vector length can be processed in a single vector instruction.
- Large memory bandwidth enabling high sustained performance.



Vector Processor on PCIe Card
(High Memory Capacity & Bandwidth Processor)

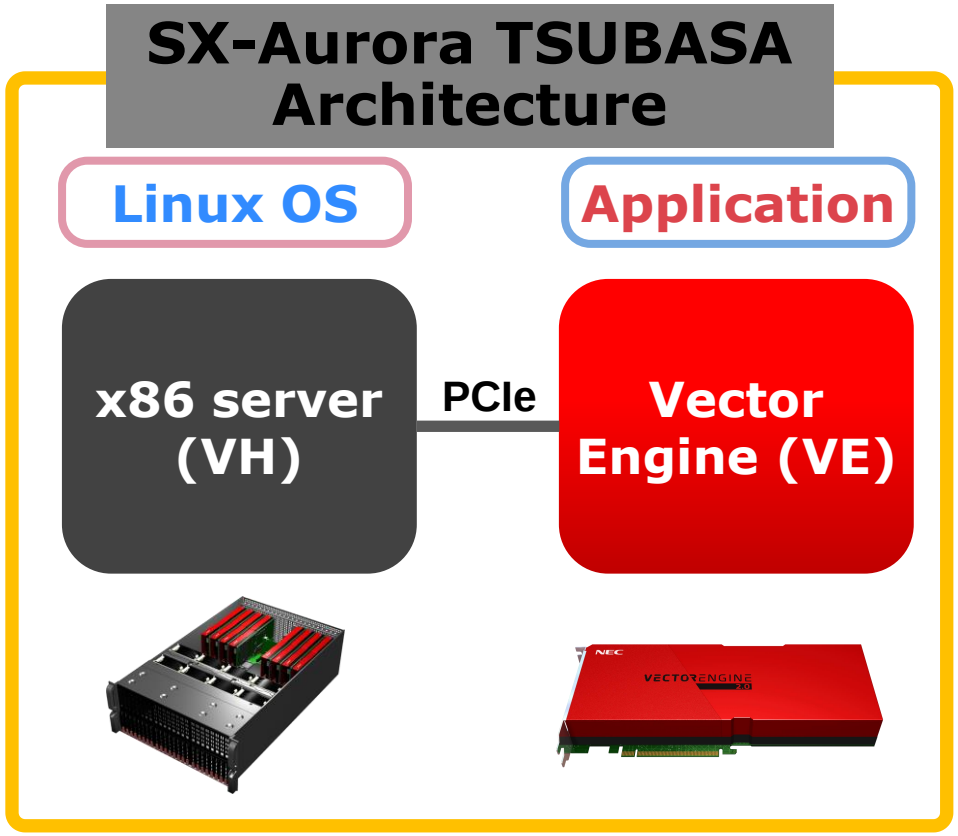


- 8 cores per processor
- 1.53TB/s memory bandwidth
- 48GB on-chip HBM2 memory
  → Very High Memory Bandwidth
- 2.45TF performance (double precision)

- Low power consumption of under 300W
  → Operational power consumption around 200W
- Standard programming with Fortran/C/C++
  → No Special Programming Model Required

System Design

→ SX-Aurora TSUBASA = x86 server + VE (Vector Engine)
→ Execute C/C++/Fortran applications as-is.



Hardware

- Vector Host (Standard x86 server) + Vector Engine

Software

- Linux OS
- C/C++/ Fortran → No special programming required
- Automatic vectorization compiler

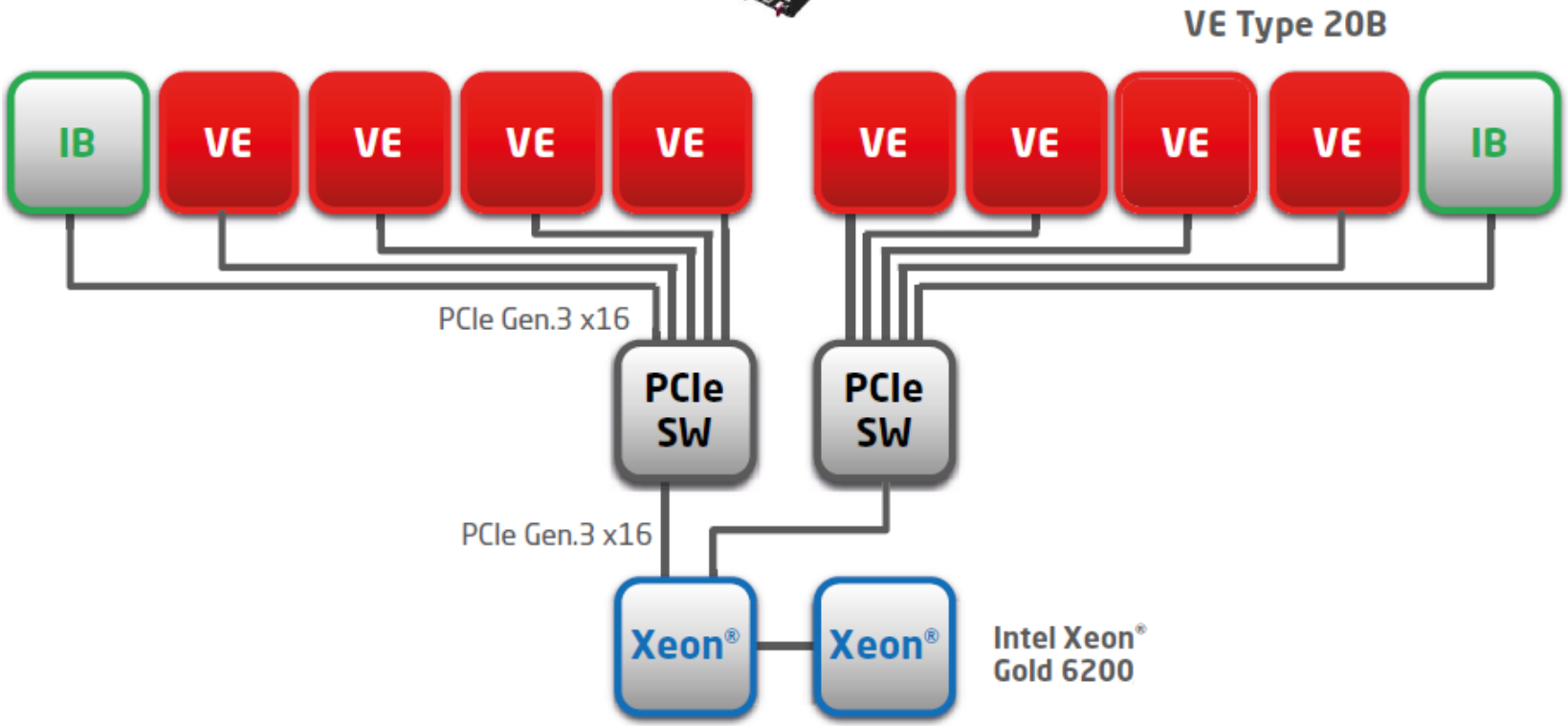
Easy programming (standard language) ▶ **Automatic vectorization compiler** ▶ **Enjoy high Performance!**

Vector Implementation of STAC-A2™ benchmark

- Greeks of an exotic American-Exercise option using the Longstaff-Schwartz early-exercise model.
- Assets, Paths and Timesteps distributed for best vectorization.
- Distributed system programmed using well known frameworks
  - Message Passing Interface (MPI)
  - OpenMP
- Target system has 8 Vector Engine cards:
  - Vector Lengths of 256 words
  - 48 GiB of HBM2 memory

System Under Test (SUT)

SX-Aurora TSUBASA B300-8



Intensive computations on large datasets

- Time-critical complex workloads to price an option, especially for large problem sizes:
  - Derivative portfolio management
  - Stock index calculations
  - Complex pricing algorithms

  … and many more such workloads that depend on deadlines for computing options

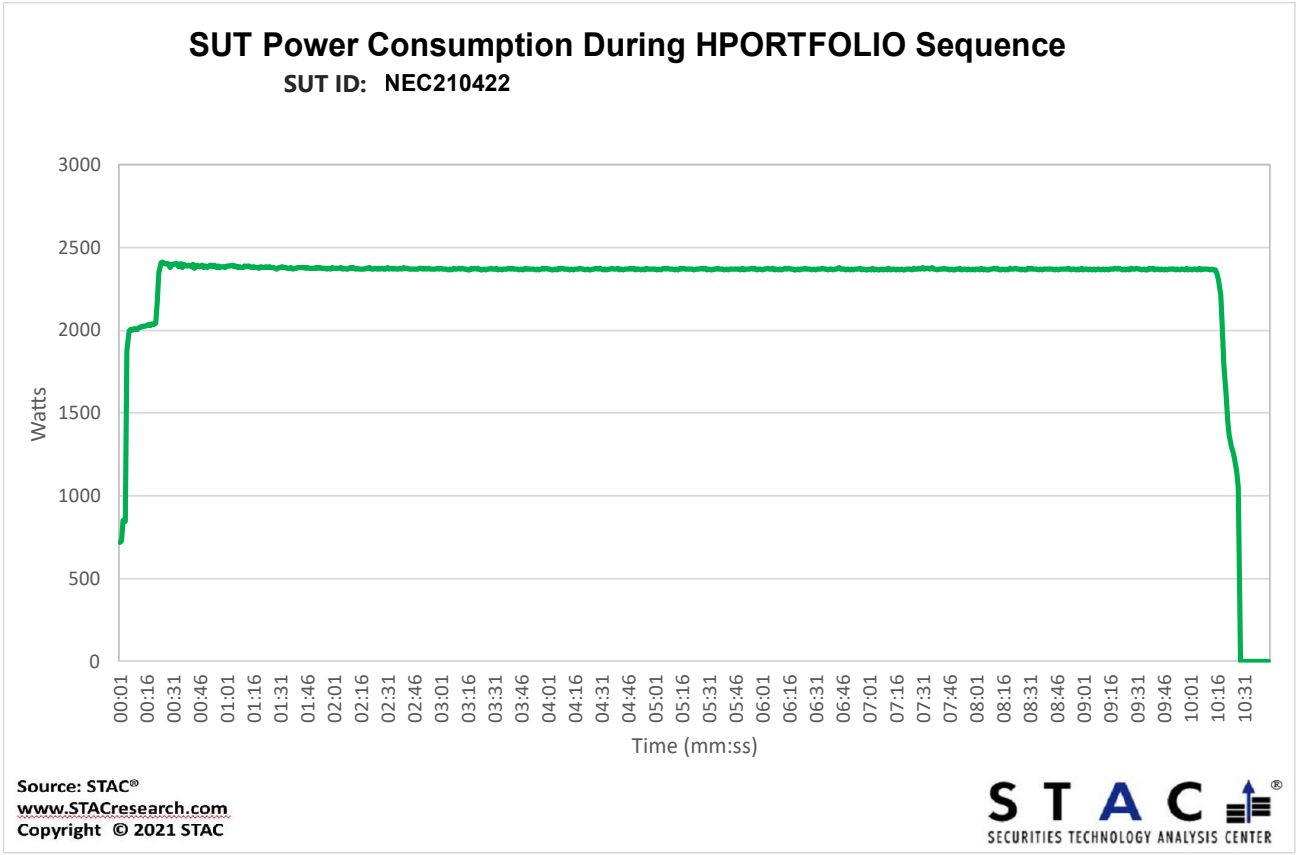
- NEC SX-Aurora TSUBASA/Vector Engine on the STAC-A2™ benchmarks:
  - Consistently high performance across options pricing algorithms
  - Set a **world record** for speed in pricing options on ***large datasets***.

    [STAC-A2.β2.GREEKS.10-100k-1260.TIME.COLD]

NEC Vector Engine is by design, the right solution for time-critical portfolio management, especially on large problem sizes!

Power efficiency for high performance options pricing

- Power efficient options pricing with high performance!
- On the STAC-A2 efficiency benchmarks:
  - under 750W when idle
  - under 2500W when fully utilized



Among the most power efficient high-performance solutions available!

Ease of programming and portability of code

The STAC-A2 Pack for NEC Vector Engine

- Developed in C.
- No special vendor-specific programming frameworks involved.
- Automatically vectorized by compiler.
- Completely portable code to any x86 environment.
- Easy to maintain.

```
$ ncc -report-format a.c -c
$ less a.L
          :
FUNCTION NAME: func
FORMAT LIST

 LINE    LOOP       STATEMENT

     5:             void func(double *x, double *y, int n )
     6:             {
     7: +-------->       for (int j = 0; j < n; j++) {
     8: |V------->           for (int i = 0; i < m; i++)
     9: |V------                 a[i] += b[i] * c[j];
    10: +------          }
    11:            }
…
```

Compiler automatically vectorizes computational loops

\Orchestrating a brighter world

NEC creates the social values of safety, security,
fairness and efficiency to promote a more sustainable world
where everyone has the chance to reach their full potential.