Fault-resilient algorithms for Exascale CFD

Eman Bagheri and Manuel Münsch

CEEC

Centre of Excellence in Exascale CFD

Failure in HPC clusters

- ECMWF (7220 Cray XC-40) with15 node failures per months excluding preventive maintenance

Hardware Failures
- Processor-Related
- Memory
- Storage
- Network Hardware
- Power Supply and Distribution
- Cooling and Environmental
- Peripheral and Component

Software Failures
- Operating System
- Resource Management and Job Scheduling
- Network Software
- Data and File System
- Application-Level Failures

Risk of Failure

- ECMWF (7220 Cray XC-40) with15 node failures per months

$$\mathbf{P(failure) = 1 - e^{-\left(\frac{CT}{MTBF} \times \frac{Nodes_{Job}}{Nodes_{Total}}\right) \times \left(1+\frac{QT}{CT}\right)}}$$

MTBF : Mean Time Between Failure
CT: Compute Time
QT: QueueTime

$Nodes_{Job}$: Number of requested nodes
$Nodes_{Total}$: Total number nodes

Assumptions:

- Exponential MTBF distribution: constant failure rate over time, independent failures
- Uniform node failure distribution
- No redundancy or failover

MTBF = 48 $\left[\frac{h}{Failure}\right]$
$Nodes_{Total}$ = 7220

Job1
CT = 10 days;
$Nodes_{Job}$ =100
QT = 1 $\left[\frac{h}{Job}\right]$

$P(failure) = 7\%$

Job2
CT = 30 days;
$Nodes_{Job}$ =1400
QT = 1 $\left[\frac{h}{Job}\right]$

$P(failure) = 77\%$

Resilience methodologies

- Checkpointing to stable storage at constant intervals
- Remote in-memory checkpointing
- Coarse resolution backup grids
- Checkpoint-restart using lossy compression
- Process replication
- MPI online rollback recovery methods
- Algorithmic resilience: Recovery-restart for sparse linear solvers

.

Dynamic checkpointing

- Many hardware failures (overheating, component wear, network connectivity issues) show progressive signs
- Failures often preceded by performance degradation or unusual behaviors
- Online monitoring of the system's performance and checkpointing when a failure is likely to happen
- Reducing I/O overheads
- No information loss
- Requires minimal overhead for system‘s monitoring

LIKWID: a performance monitoring tool

likwid-powermeter : Measure energy consumption a temperature

- Online monitoring of core temperature
- Accesses RAPL counters on Intel processo
- Predict if core-overheating is likely

```
--------------------------------------------------------------
CPU name:       Intel(R) Xeon(R) Platinum 8360Y CPU @ 2.40GHz
CPU type:       Intel Icelake SP processor
CPU clock:      2.39 GHz
--------------------------------------------------------------
--------------------------------------------------------------
Runtime: 2.00009 s
Measure for socket 0 on CPU 0
Domain PKG:
Energy consumed: 256.168 Joules
Power consumed: 128.078 Watt
Domain PP0:
Energy consumed: 0 Joules
Power consumed: 0 Watt
Domain DRAM:
Energy consumed: 21.7544 Joules
Power consumed: 10.8767 Watt
Domain PLATFORM:
Energy consumed: 0 Joules
Power consumed: 0 Watt

Measure for socket 1 on CPU 36
Domain PKG:
Energy consumed: 261.349 Joules
Power consumed: 130.669 Watt
Domain PP0:
Energy consumed: 0 Joules
Power consumed: 0 Watt
Domain DRAM:
Energy consumed: 31.382 Joules
Power consumed: 15.6903 Watt
Domain PLATFORM:
Energy consumed: 0 Joules
Power consumed: 0 Watt
--------------------------------------------------------------
```

```
------------------------------------
CPU name:       Intel(R) Xeon(R)
CPU type:       Intel Icelake SP
CPU clock:      2.39 GHz
------------------------------------
------------------------------------
Current HW thread temperatures:
Socket 0 HWThread 0: 37 C
Socket 0 HWThread 1: 35 C
Socket 0 HWThread 2: 39 C
Socket 0 HWThread 3: 38 C
Socket 0 HWThread 4: 37 C
Socket 0 HWThread 5: 37 C
Socket 0 HWThread 6: 40 C
Socket 0 HWThread 7: 37 C
Socket 0 HWThread 8: 38 C
Socket 0 HWThread 9: 39 C
Socket 0 HWThread 10: 38 C
```

LIKWID: a performance monitoring tool

likwid-perfctr - : A tool for accessing hardware performance counte

- Simple end-to-end measurement of hardware performance metri

- Offers various measurement groups

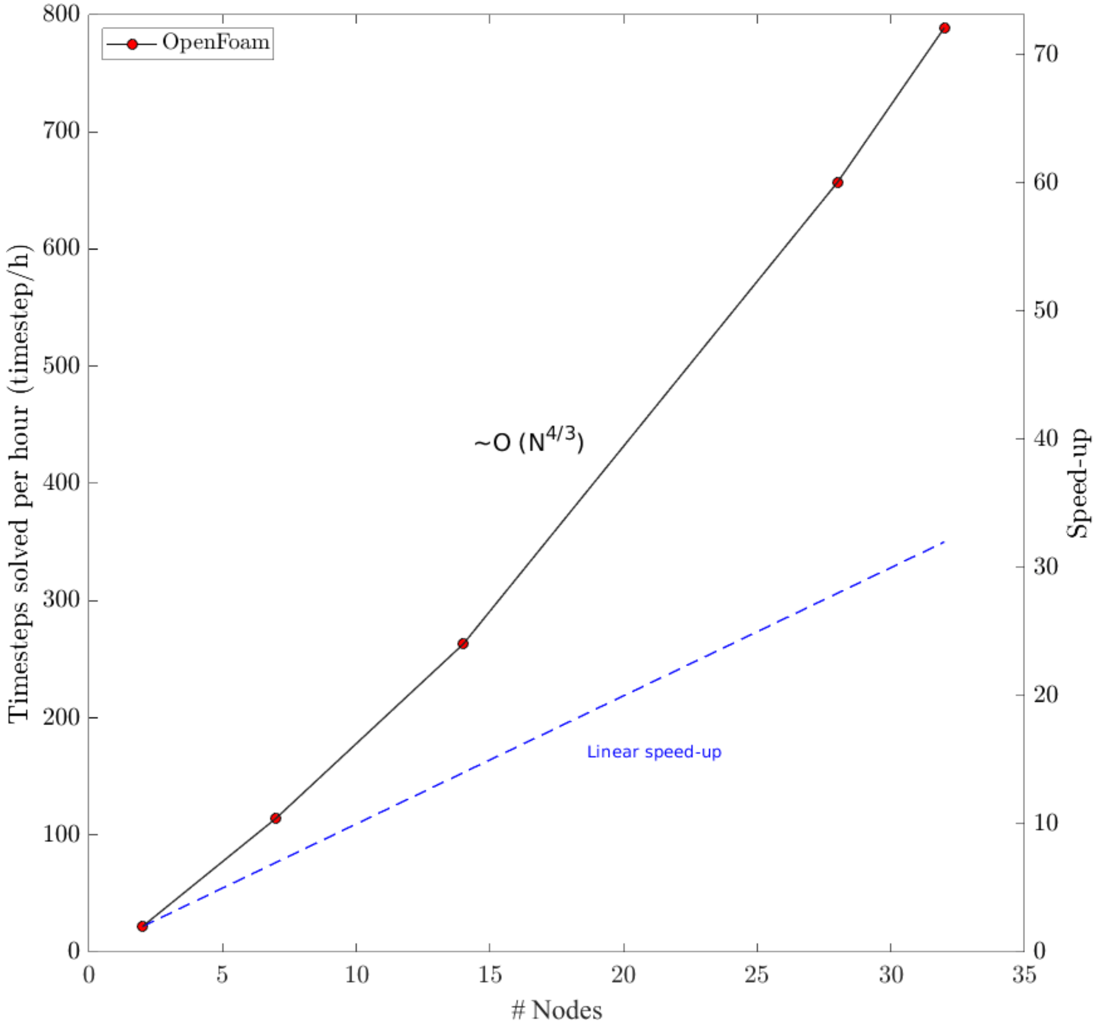
- Supports: x86, ARM, POWER CPUs, Nvidia co-processors.

- Operating modes:
  - Wrapper
  - Stethoscope
  - Timeline
  - Marker API

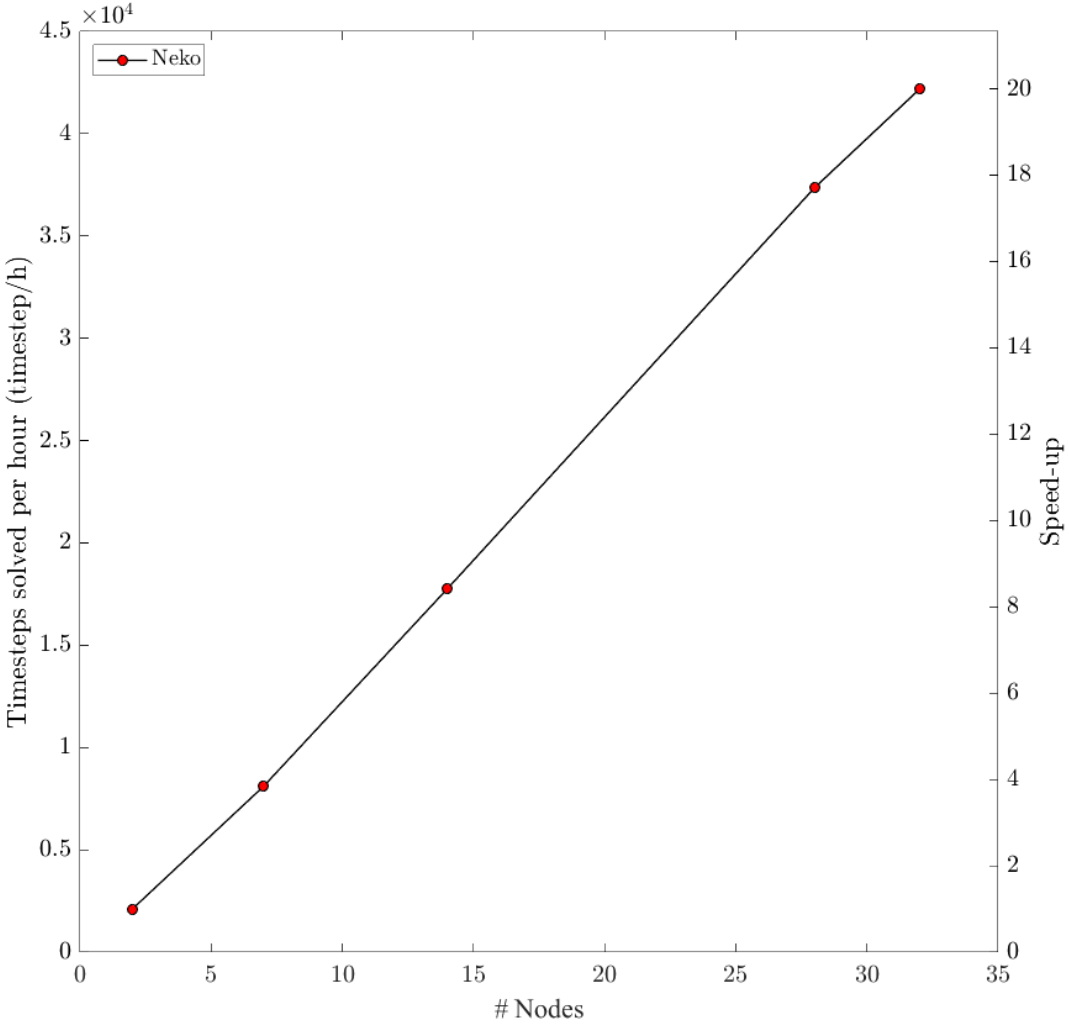
Group name	Description
TMA	Top down cycle allocation
MEM_FREERUN	Memory bandwidth in MBytes/s
MEM	Memory bandwidth in MBytes/s
L2	L2 cache bandwidth in MBytes/s
BRANCH	Branch prediction miss rate/ratio
DIVIDE	Divide unit information
MEM_DP	Overview of arithmetic and main memory performance
MEM_SP	Overview of arithmetic and main memory performance
L2CACHE	L2 cache miss rate/ratio
FLOPS_SP	Single Precision MFLOP/s
L3	L3 cache bandwidth in MBytes/s
CYCLE_STALLS	Cycle Activities (Stalls)
FLOPS_DP	Double Precision MFLOP/s
FLOPS_AVX	Packed AVX MFLOP/s
DATA	Load to store ratio
ENERGY	Power and Energy consumption
UPI	UPI data traffic
CLOCK	Power and Energy consumption
CYCLE_ACTIVITY	Cycle Activities

- Marker API can cause overhead
- **Stethoscope** mode allows you to “listen” to what is currently happening, without any overhead

likwid-perfctr stethoscope mode

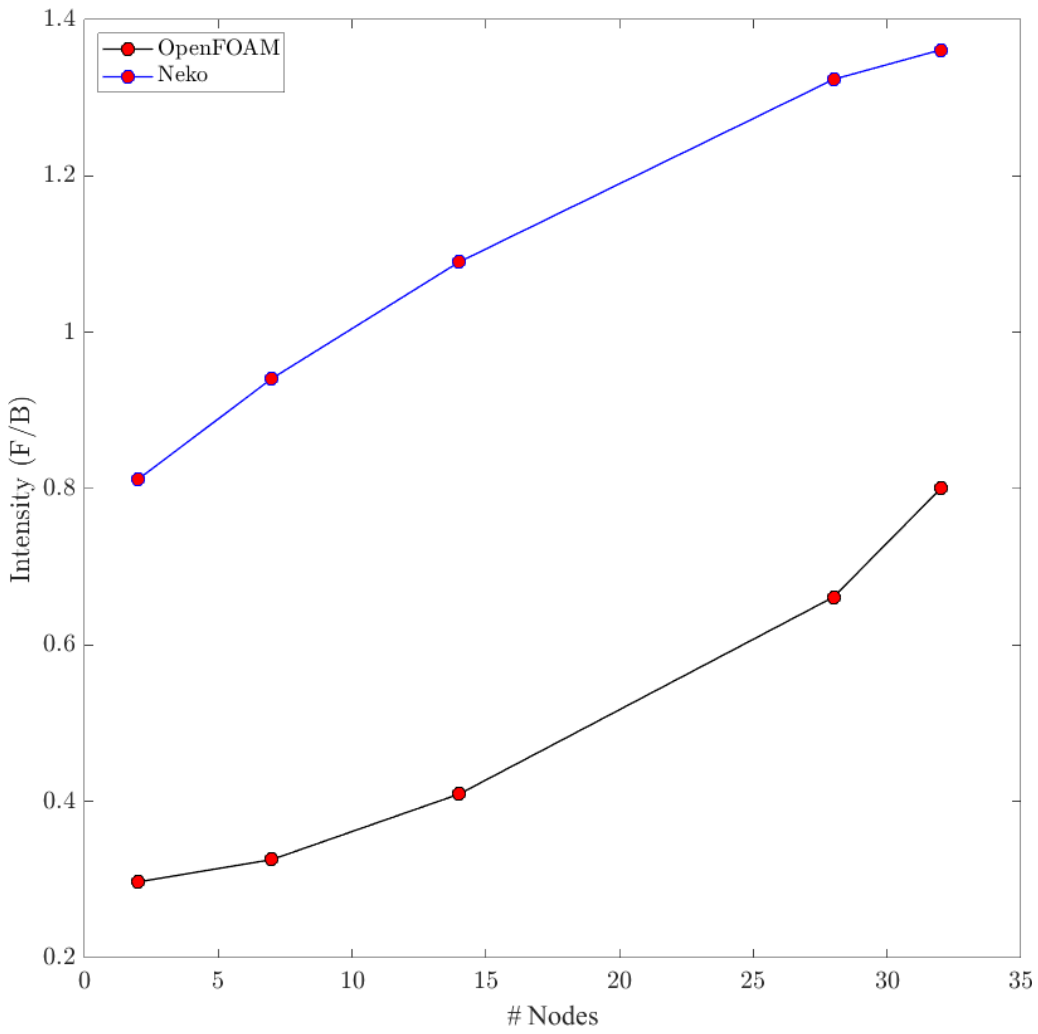


OpenFOAM: (84 million cells)

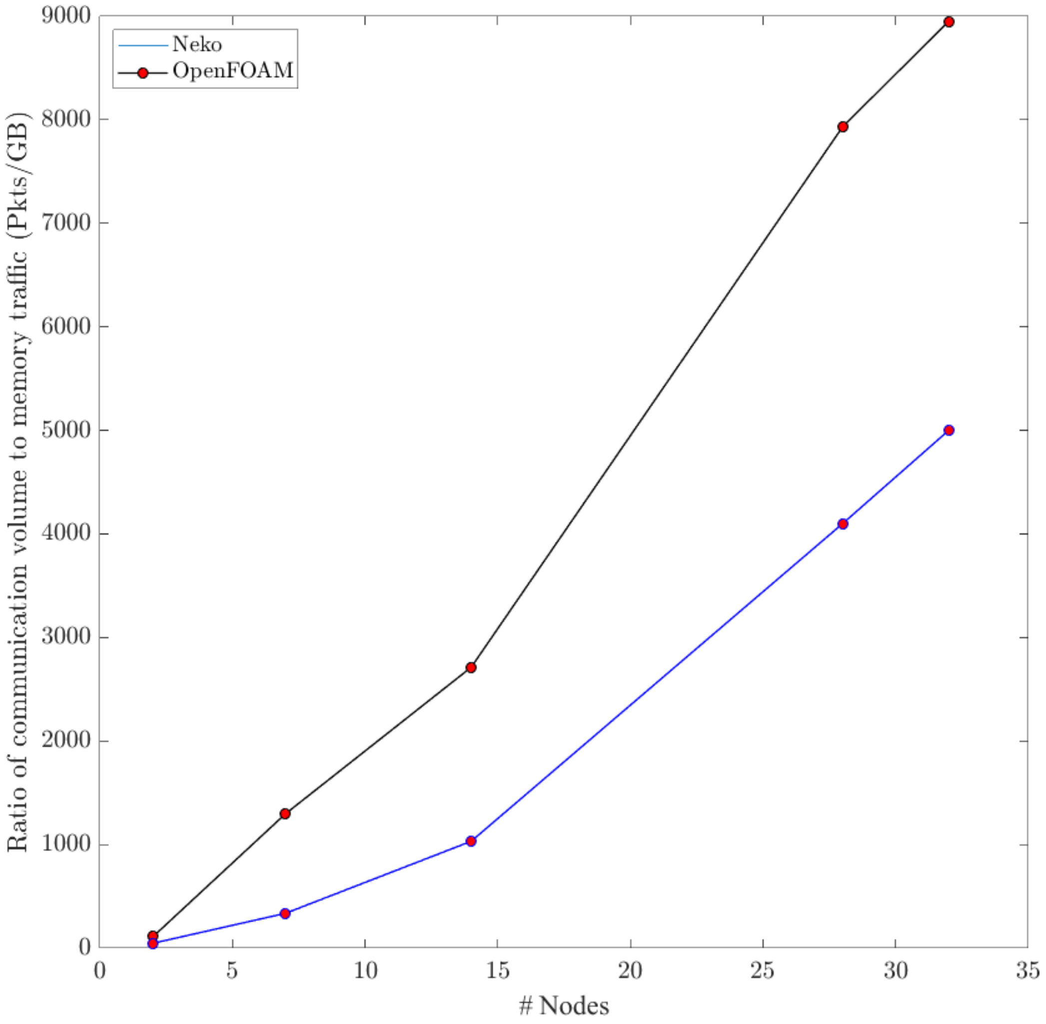


Neko: (89 million GLL Points)

likwid-perfctr stethoscope mode



Computational Intensity



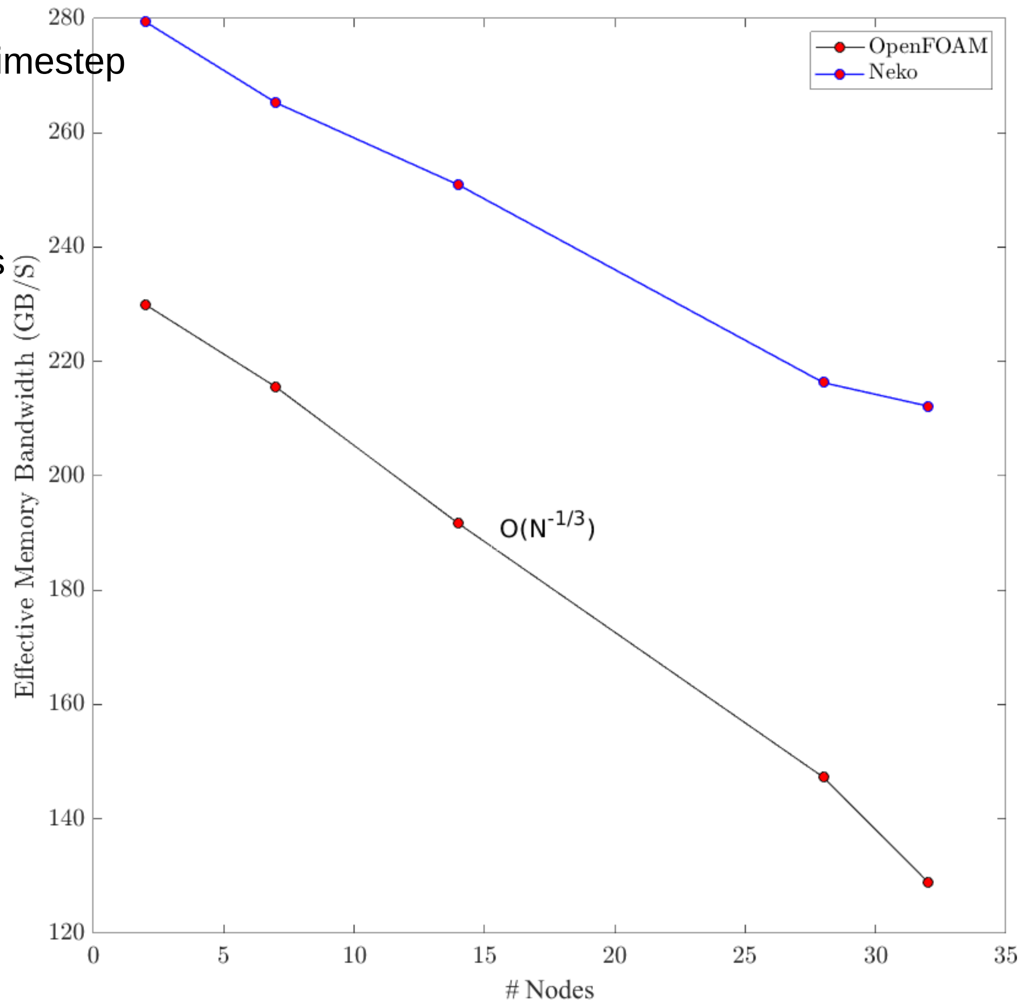
Packets/GB

likwid-perfctr stethoscope mode

- L^3: Total number of elements to be transferred from the main memory per physical timestep
- α: Size of each element (Byte)
- N: Number of compute nodes
- $T(N)$: Time (s) to transfer data from the main memory when parallelized on N nodes
- $V_c(N)$: Communication volume (Byte) due to halo layers
- λ: Network latency (s)
- b_m: Memory bandwidth (Byte/s)
- b_{net}: Network bandwidth (Byte/s)
- b_{eff}: effective bandwidth (Byte/s)

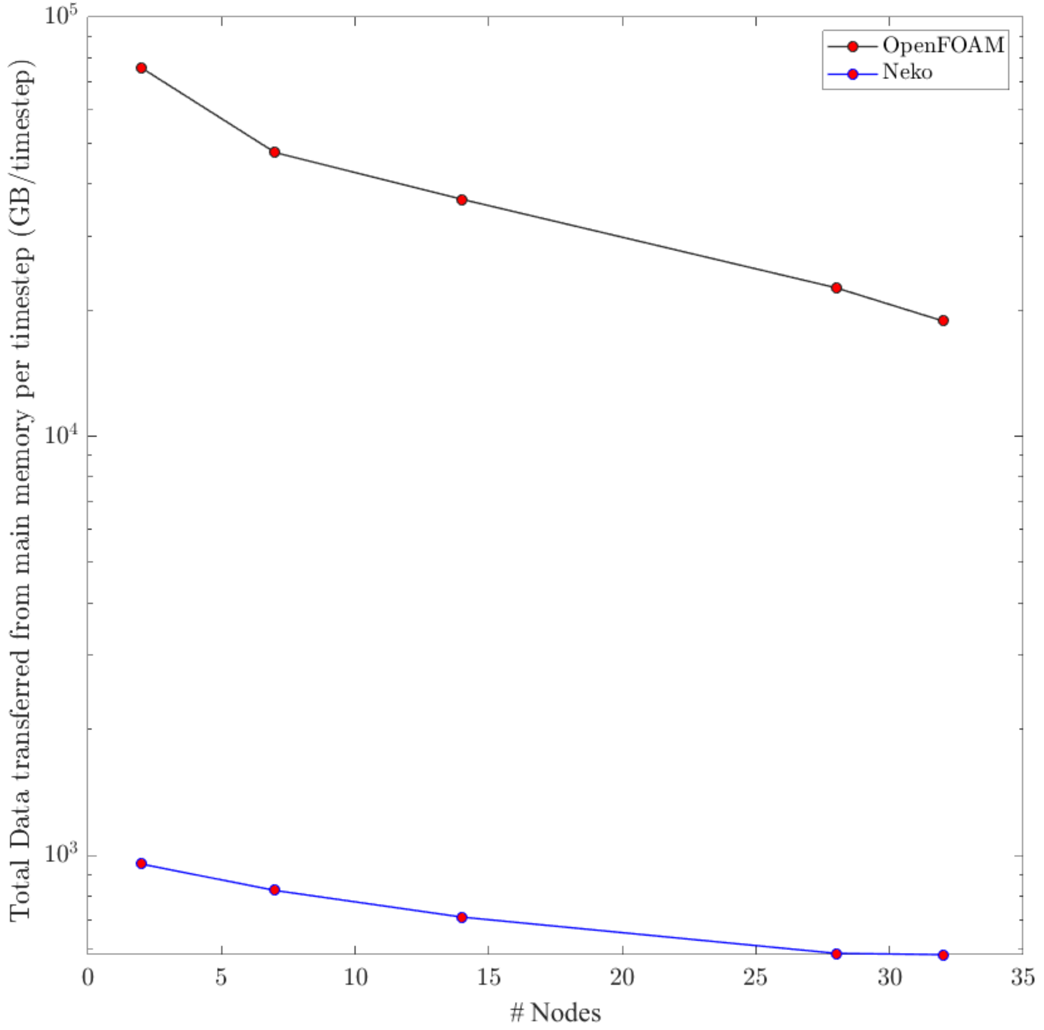
$$V_c(N) = \frac{\alpha L^2}{N^{2/3}} \quad V_d(N) = \frac{\alpha L^3}{N}$$

$$b_{eff}(N) = \frac{V_d(N)}{T(N)} = \frac{b_m}{1 + \frac{1}{L}\frac{b_m}{b_n} N^{\frac{1}{3}} + \bar{\lambda} N}$$

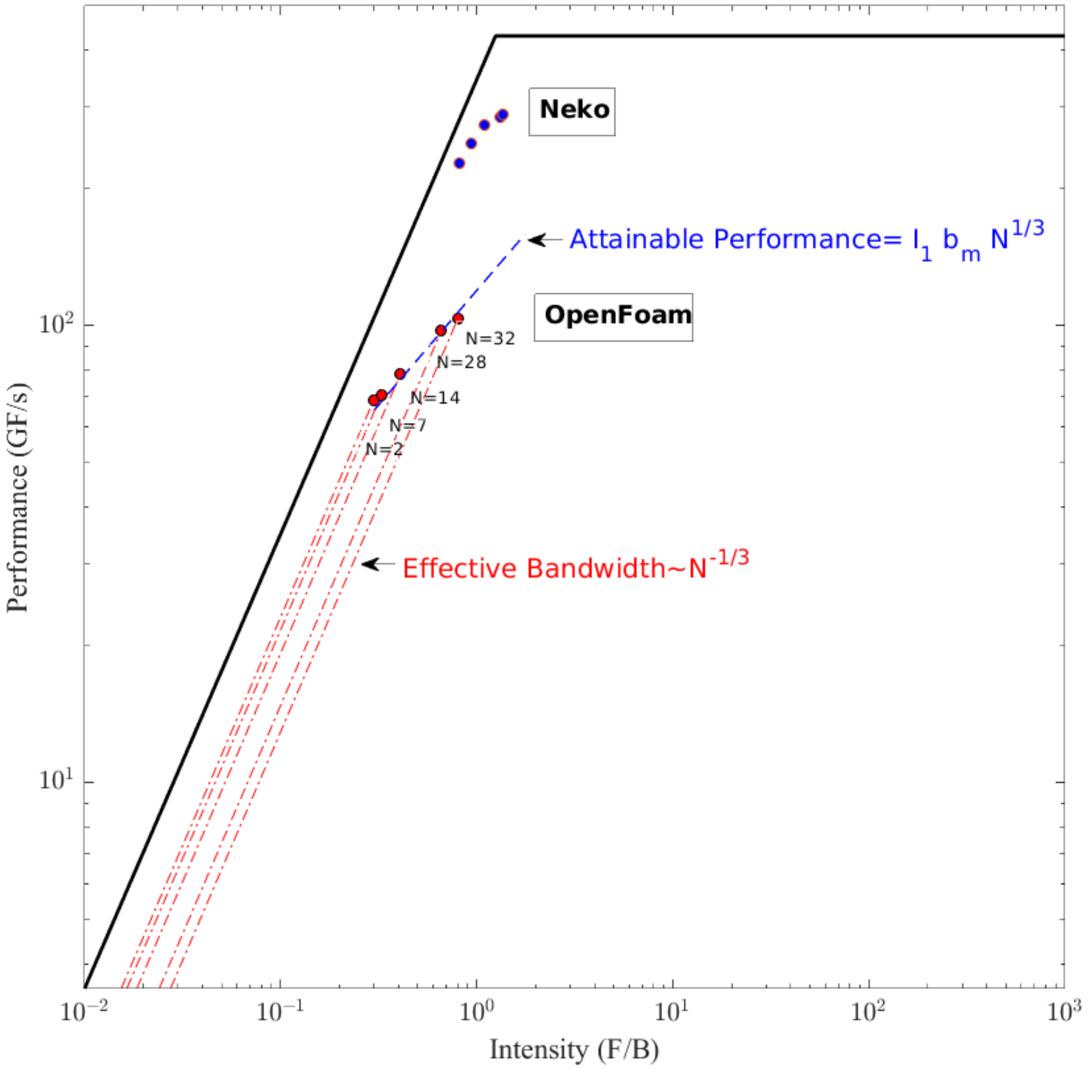


Effective Bandwidth

likwid-perfctr stethoscope mode



Data Transfer Per Timestep (GB/timestep)



Roofline Diagram

Dynamic checkpointing: Methodology

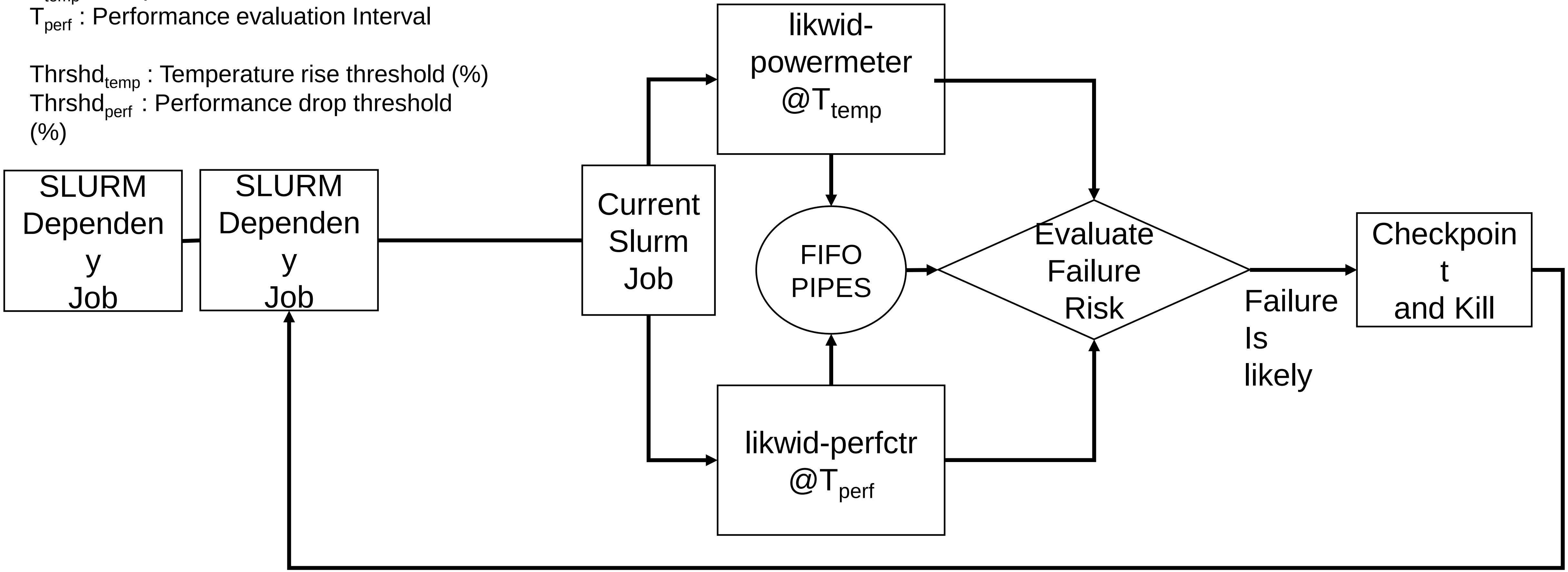
- Periodic measurements of temperature using **likwid-powermeter**
- Continuous measurements of FLOP/s using **likwid-perfctr**
- Upstream slow-downs are reflected in FLOP/s drops
- Obtain expected performance indicators
- Failure risk evaluation
- Checkpointing in case of persisting anomalies

.

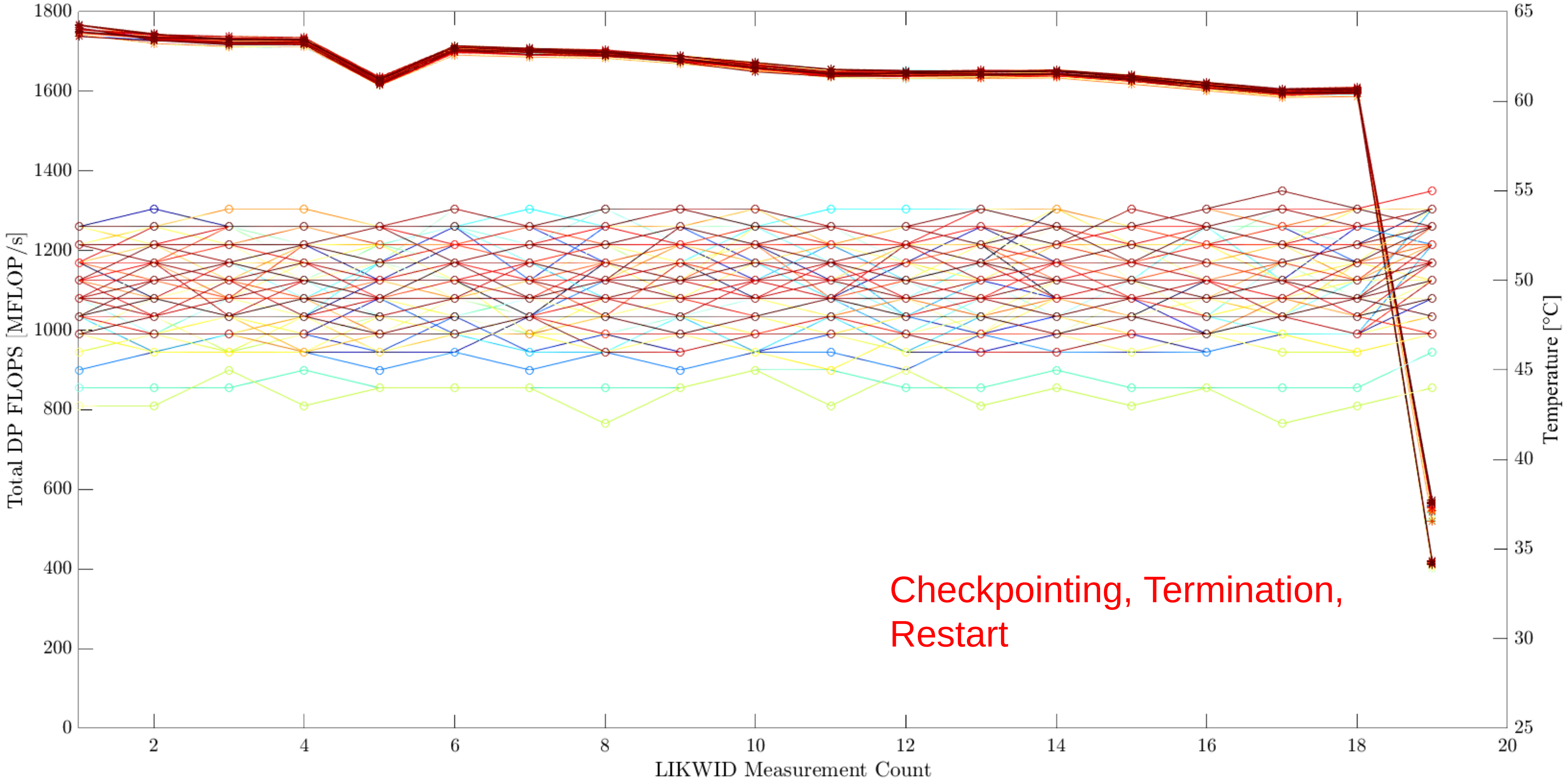
Dynamic checkpointing: Methodology

Inputs:
T_{temp} : Temperature Measurement Interval
T_{perf} : Performance evaluation Interval

$Thrshd_{temp}$: Temperature rise threshold (%)
$Thrshd_{perf}$: Performance drop threshold (%)



Dynamic checkpointing: Initial tests



Dynamic checkpointing

Advantages:

- Highly efficient with no noticeable overhead
- Effective in case of a slowdown in any network components
- Runtime information about performance and load imbalance

Disadvantages :

- No mechanism to handle soft faults such as bit-flips
- No mechanism to handle sudden hardware faults like power outage physical damage
- More careful monitoring of temporary performance drops
- User-dependent input thresholds

.

Summary

- Fault resillient algorithms are necessary for Exascale simulations
- Likwid provides effiecient tools for performance monitoring
- Dynamic checkpointing is an effective and efficient non-intrusive method to handle gradual failures
- Provide performance metrics for no additional overhead
- Future work to extend the method for GPUs and to detect temporary performance drop more elaborately

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Sweden, Germany, Spain, Greece, and Denmark under grant agreement No 101093393.

Co-funded by the European Union

EuroHPC Joint Undertaking

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European High Performance Computing Joint Undertaking (JU) and Sweden, Germany, Spain, Greece, and Denmark. Neither the European Union nor the granting authority can be held responsible for them.

CEEC

Centre of Excellence in Exascale CFD