# Have you ever seen the … noise?

Jesus Labarta, BSC

EU H2020 Centre of Excellence (CoE)

1 December 2018 – 30 November 2021

Grant Agreement No 824080

# Objective

- Highlight the importance of understanding how the FOA trace we analyze may have been impacted by perturbations (“noise”) and build confidence in our performance assessments

- We will describe/demonstrate:
  - how to use Paraver to “identify” noise
  - Develop some perception of its potential impact
  - On a detailed FOA trace & on a filtered trace

# Noise

- Variability that we do not understand → **Noise**
  - Perturbations, interference on our application from other activities that may be sharing our resources
- Noise is a property of nature → try to understand, learn how to live with it.

- Some level of fine grain observation and quantification of its effect:
  - Will always be approximate. In the end it is noise.
  - Will always be useful to complement our observations and comments as analysts on the behavior of applications. Try to separate blame.

# How

- **Look**ing for characteristic patterns on full, filtered or burst mode traces
- **Quantify**ing the impact on traces containing cycles and instruction hardware counters
- **What if** (noise was not there) predictions

# (Views, simulations) and noise

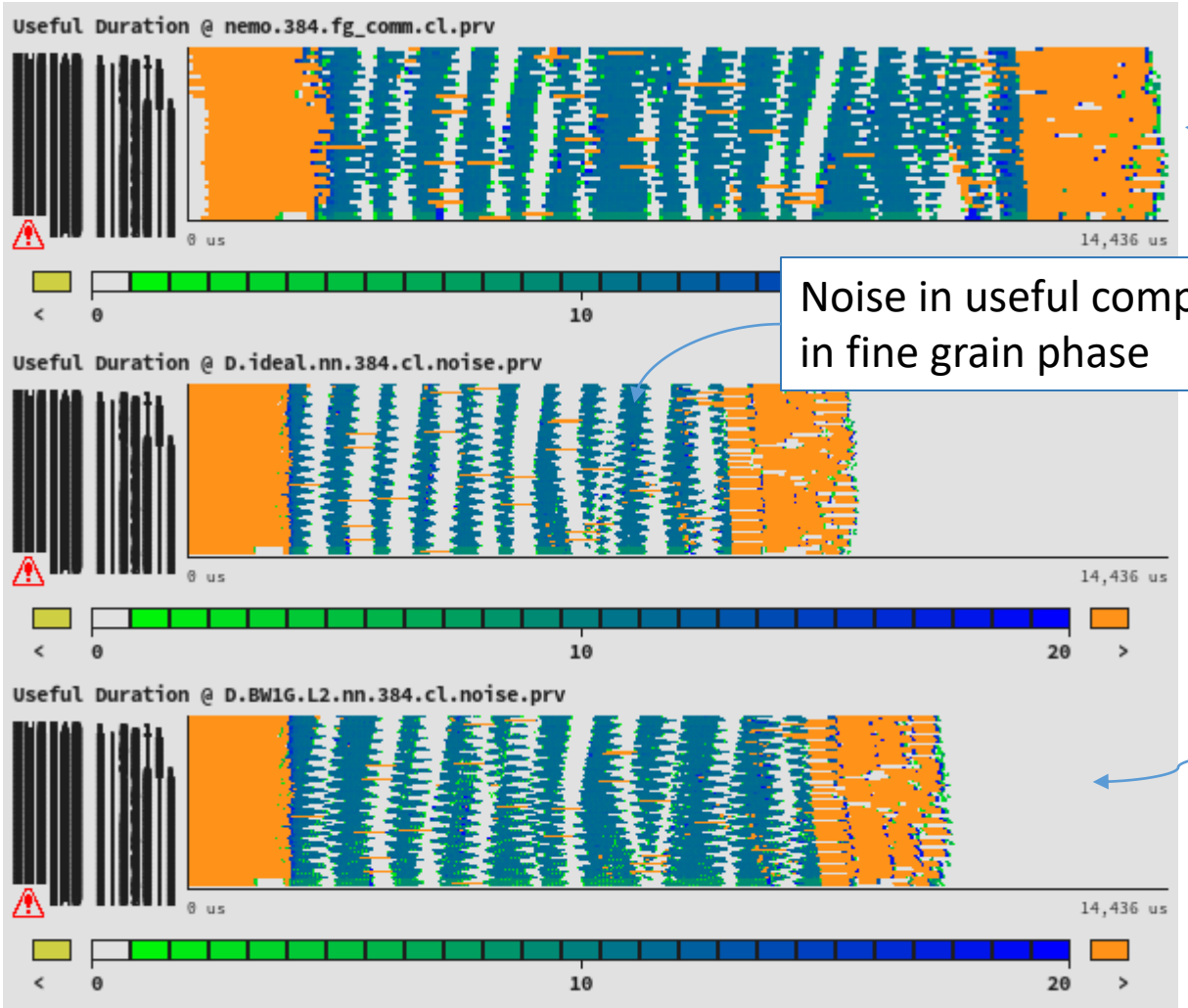
- Visually identify patterns and speculate on causes and effects
  - Useful duration view, histograms of useful duration, MPI calls duration, …
  - Random/scattered variabilities
- Quantify presence
  - Effective frequency (Cycles per microsecond)
  - Preempted time
- Predict absence
  - Dimemas predicted behavior (with ideal or nominal network) eliminates noise in communications
  - Eliminating the impact of noise in user level computation
  - From extremely perturbed traces (oversubscribed runs) → scalability predictions

# Noisy patterns

Actual run

Ideal Network

“Nominal” Network




Noise in useful computation in fine grain phase

Noise ?

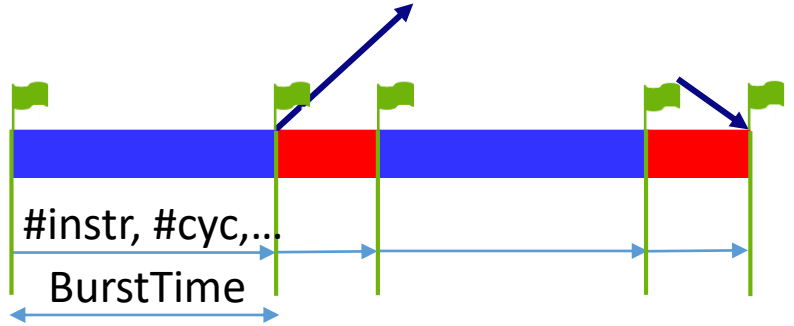
Randomness in patterns

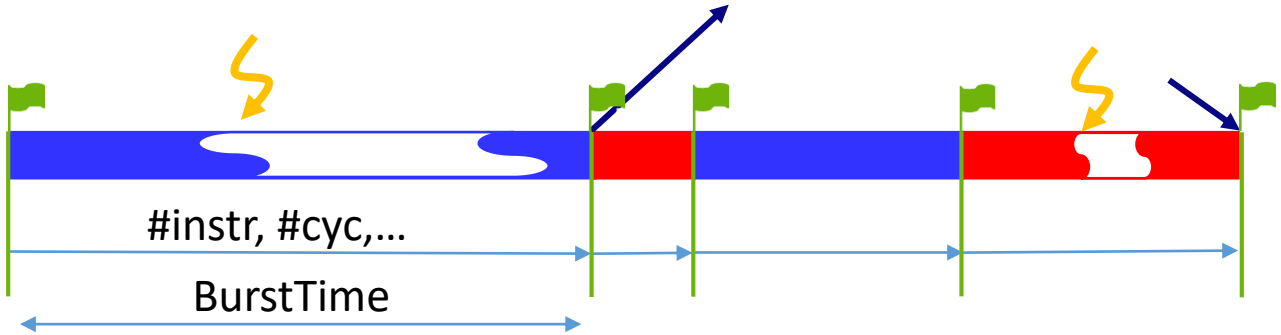
Difference → there was noise in the internals of MPI implementation

Dimemas does not have noise
Check duration of MPI calls

# Quantifying noise

- Good old times where frequency was known are gone
  - Turbo
  - DVFS
  - Power capping, governors
  - Device variability
- PAPI counters virtualized
  - Kernel calls, yields
  - Preemptions




$$Freq = \frac{\#cyc}{BurstTime}$$

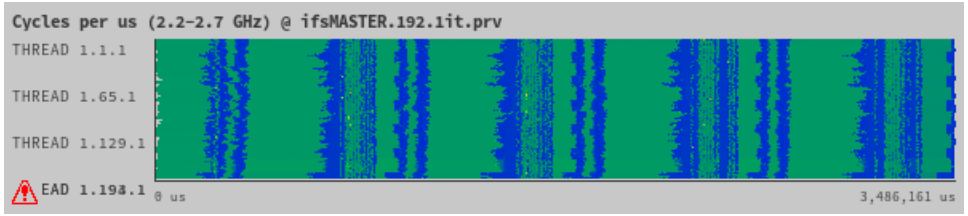
# Only noise?

- Low effective frequency also results when OS kernel functions are invoked by the application even if the process does not context switch
  - I/O
  - Memory management

- Or not even that
  - Exceptions
  - AVX
  - …

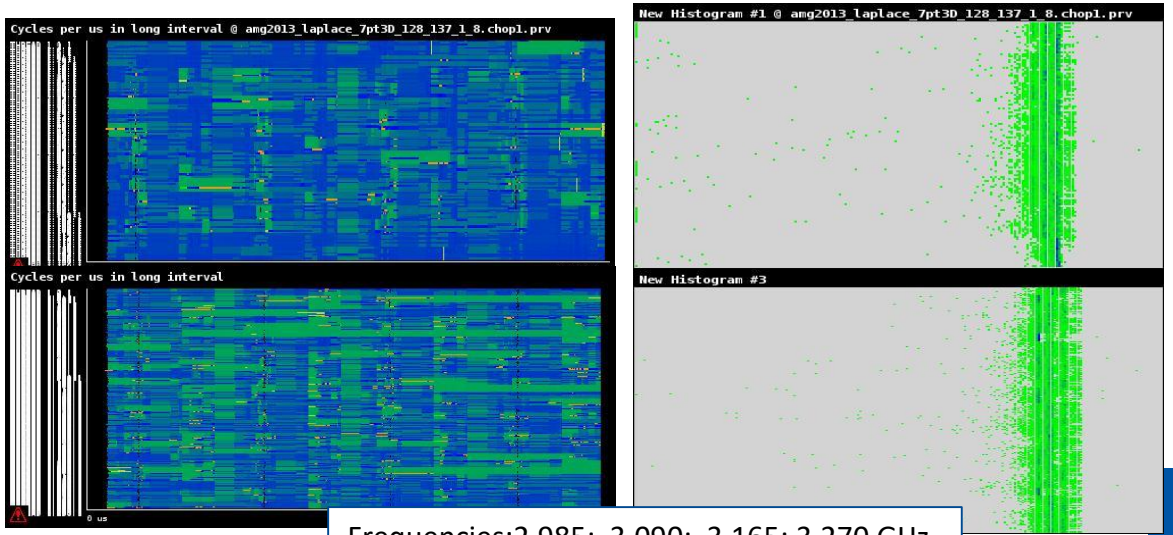
# Effective frequency changes

- Good old times where frequency was known are gone
  - Turbo
  - DVFS
  - Power capping, governors
  - Device variability




Towards unpredictable core performance

- PAPI counters virtualized
  - Kernel calls, yields
  - Preemptions



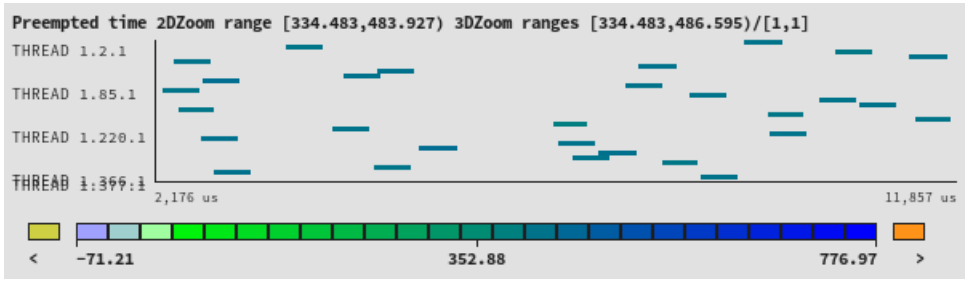

Frequencies:2.985; 3.090; 3.165; 3.270 GHz

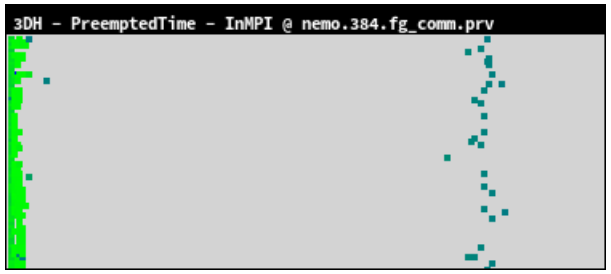
# Have you ever seen the noise?

"Preemptions"

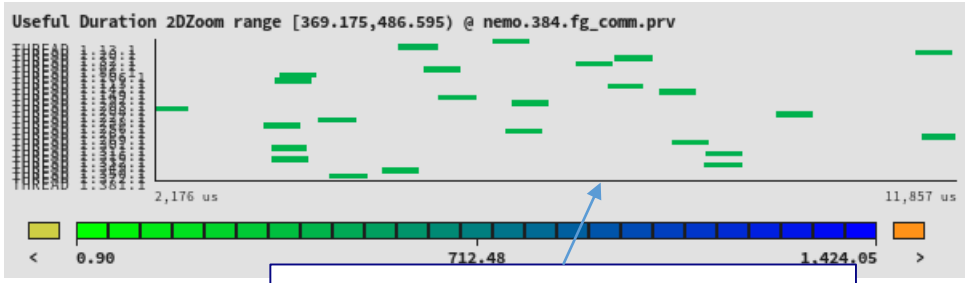
"Preempted" time

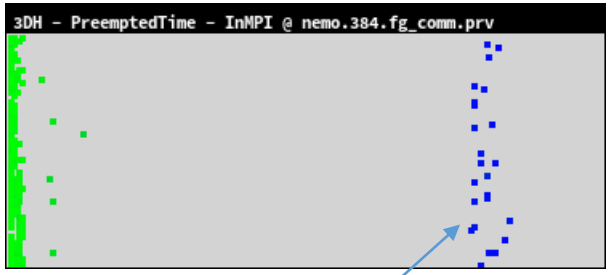
In MPI




In Useful




Scattered in space and time
Both in MPI and useful

Mode ~400 us

"Noise" cause ?
Cant fight noise, learn to live with it

# Noise and the efficiency model?

- Can be reflected in:
  - Load balance: if few instances or concentrated in few processes and long
  - Serialization: large number of instances during the FOA, uniformly distributed among processes
  - Transfer: the preemptions fall within the MPI calls

# Performance Optimisation and Productivity

A Centre of Excellence in HPC

## Contact:

- https://www.pop-coe.eu
- pop@bsc.es
- @POP_HPC
- youtube.com/POPHPC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 676553 and 824080.