

## PhD Meeting 8 Sep 2017

### Discussing last meeting's action points

- single vs double precision was not checked
- not an easy task to accomplish in few weeks
- a dirty estimation would be sufficient
- There are libraries for custom precision
- approximate computing is a hot topic for CS
  - custom precision to minimize power
- We need sudo rights to access power consumption counters etc
- **AP** Konstantinos check with the system admin what we can do
- our matrices are not sparse, we don't need to optimize them for memory usage
- idea for a histogram with a variable width slicing is interesting

### Konstantinos presentation

- Overview of Blond main functions and datastructures
- test-cases profiling:
  - LHC
    - Not doing well in scalability
    - Some functions need to be optimized
    - This could be a nice story for a paper
    - Thread interference should be considered
  - PSB
    - Scaling good enough (42x theoretical peak speedup)
    - amdahl's law is not the best approximation as it doesn't consider memory BW
    - Roofline model is a better approximation
      - Computational intensity
      - memory BW
    - Roofline model has not been yet studied for multi-node machines
- Code optimizations
  - Histogram:
    - loop tiling → better cache usage + vectorization
    - speculation
    - 20-25% speedup
  - linear interpolation:
    - loop tiling
    - precalculating a part of the loop to save some computations
    - 35-40% speedup
  - in a paper I will have to prove that the threads to CPUs configuration is the optimal
  - numa\_ctl to control thread configuration and memory allocations
  - **AP** Overlapping kick and drift would improve the memory usage
- python PAPI library → let us extract processor counters and metrics, can be useful in the future
- ISCAS
  - Deadline: October 16
  - Size: 4 Pages (only)

- Contents:
  - coarse analysis of the BLoND algorithm
  - Explain how BlonD utilizes heavily DSP and signal processing
  - Optimizations: mutli-threading + vectorization only
  - **AP** integrate the GPU parts in the code
  - Considering only the OFTB module might be better for the size of the paper
- next meeting 29<sup>th</sup> September