Using the Delft Blue supercomputer for processing massive point clouds

Nauman Ahmed

Research Software Engineer

netherlands eScience center

netherlands eScience center

Delftblue supercomputer

Node Category	Number	Cores	CPU/GPU	RAM	SSD
Standard	218	48	2x Intel XEON E5-6248R 24C 3.0GHz	192 GB	480 GB
Fat type-a	6	48	2x Intel XEON E5-6248R 24C 3.0GHz	768 GB	480 GB
Fat type-b	4	48	2x Intel XEON E5-6248R 24C 3.0GHz	1536 GB	480 GB
GPU	10	48	2x AMD EPYC 7402 24C 2.80 GHz, 4x NVIDIA Tesla V100S	32GB	256 GB

netherlands eScience center

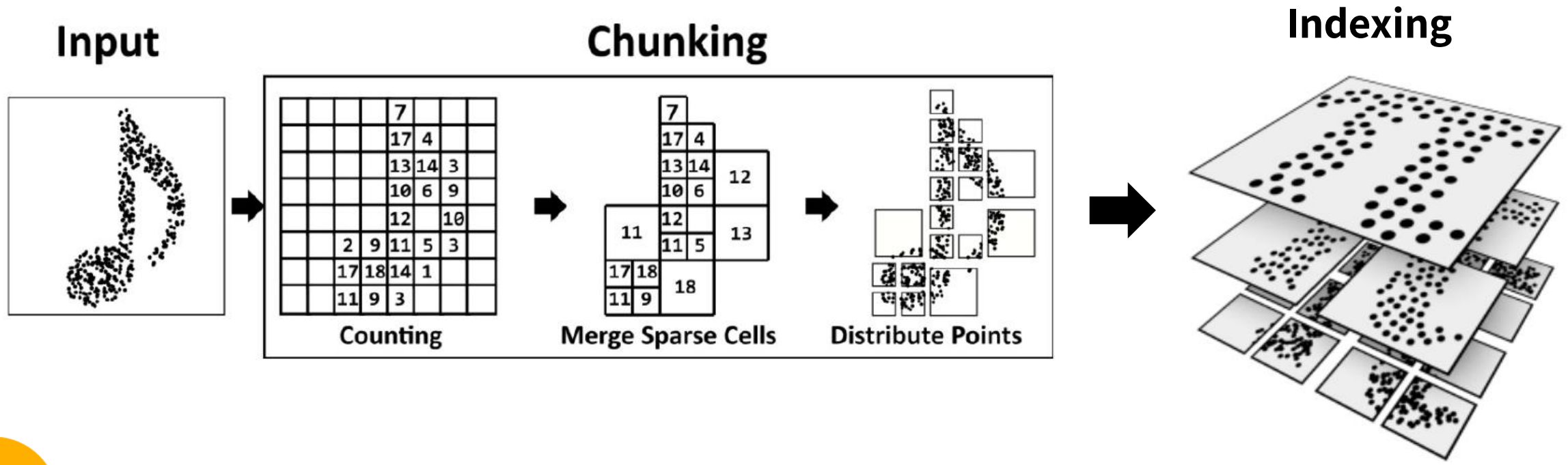
Delftblue supercomputer

- 100 Gbps interconnect between nodes
- 5 TB of storage (scratch memory) with throughput of atleast 20 GB/sec
- 200 TB of TU Delft storage connected to scratch memory via a slow connection

netherlands eScience center

PotreeConverter2

- To convert LAZ to Octree



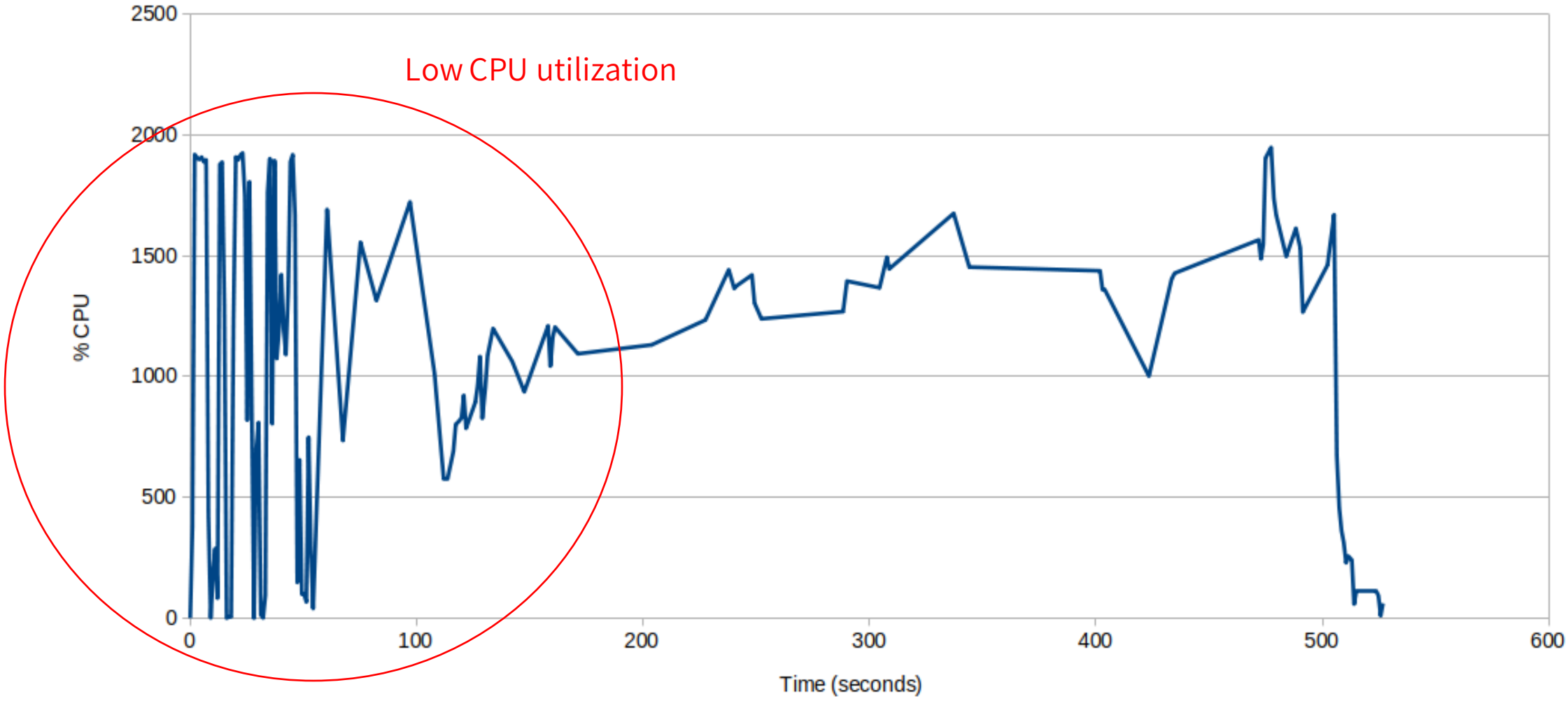
netherlands eScience center

PotreeConverter2 on Delftblue

- AHN3
  - 2. 33 TB on 1375 LAZ files
  - Data first needed to be loaded in scratch storage for processing
- The performance of PotreeConverter 2 is measured
  - with 4 AHN3 LAZ files. total points > 1 billion, size ~ 5 GB
  - 12 threads on a standard Delftblue node
  - sampling method: poisson
  - Compression: BROTLI
  - Took 11 minutes
  - Output is proprietary format
  - For whole AHN3 it will take ~ 4 - 5 days depending upon the load

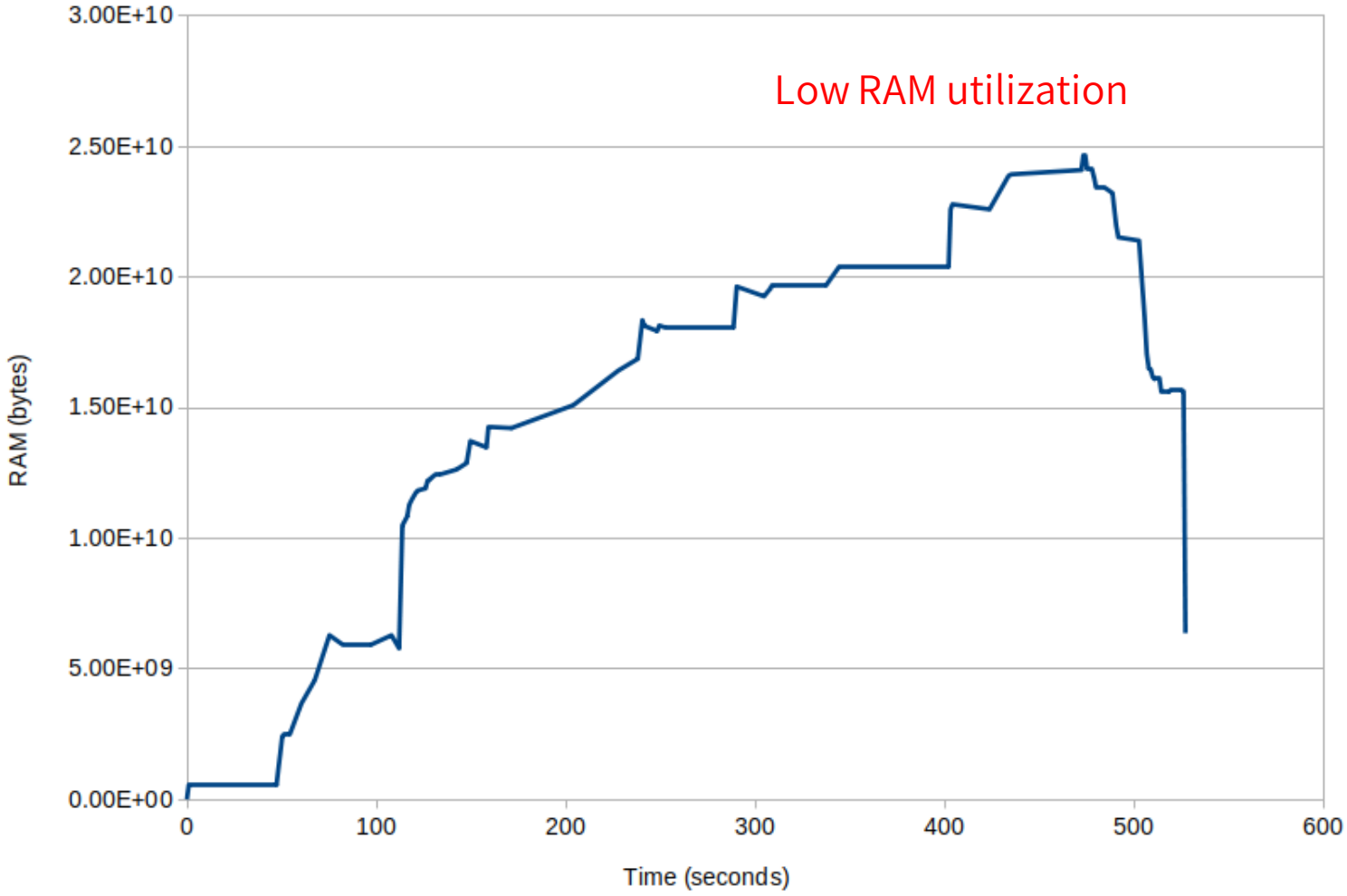
netherlands eScience center

CPU utilization



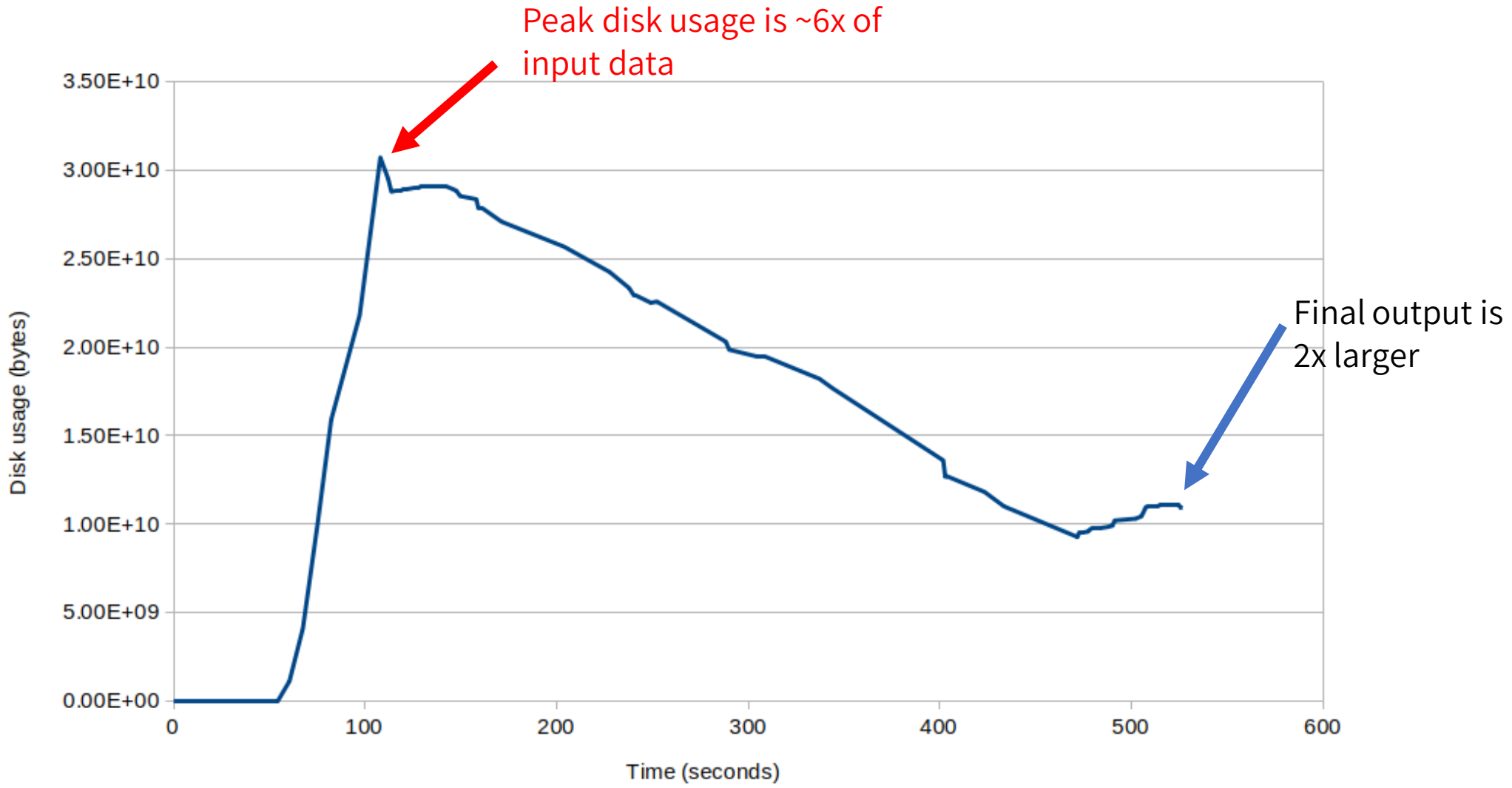
netherlands eScience center

RAM utilization



netherlands eScience center

Disk utilization



netherlands eScience center

CPU and RAM utilization issues

- Low CPU utilization in chunking and distribution due to I/O.
  - However, they make up only 25% of total runtime
- Low RAM utilization
  - Fully utilize the available RAM by memory mapping output files that are use in next stages, instead of writing them to disk. May use Fat-type nodes
- 75% time spent in indexing.
  - 70% in sampling which is based on std::sort.
- Single node performance has little room for improvement.
  - Avoid writing to shared grid in counting
  - Reducing I/O in distribution and indexing.
- MPI is used to scale the application

netherlands eScience center

Disk utilization Issues

- Peak disk utilization is about 6x larger than input data
- Can't do 2.4 TB of AHN3 in one go with 5 TB scratch storage
- Four rounds are needed i.e.
  - split the input data in 4 parts.
  - Run PotreeConverter2 on each part till "indexing"and save the output to TU Delft storage
  - Load the hierarchy information of all four parts and perform the final merging.

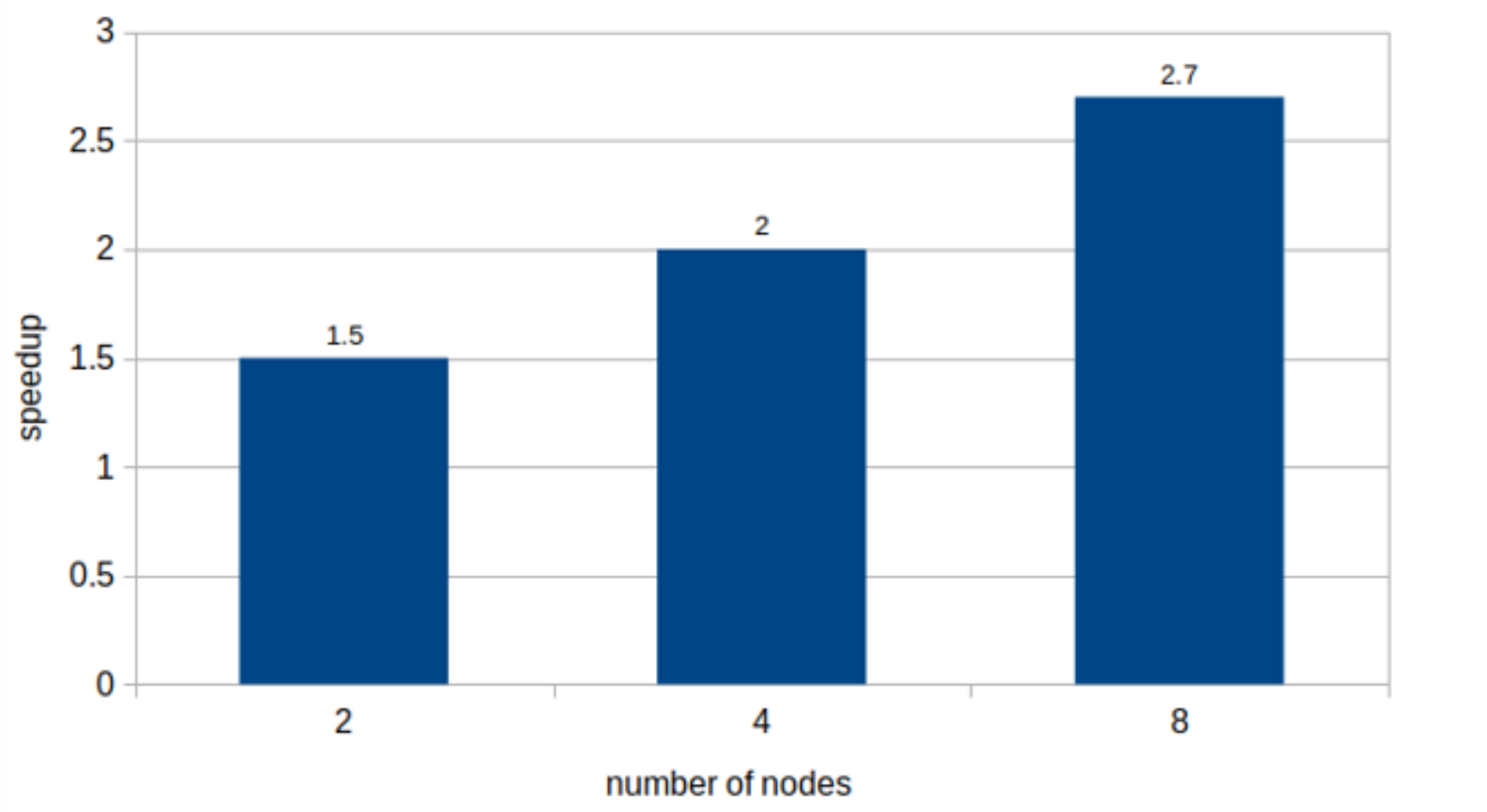
netherlands eScience center

PotreeConverter2 MPI

- In indexing all chunks are processed independently
- Indexing on multiple nodes. Maximum speedup ~4x
- Input and settings
  - with 20 AHN3 LAZ files. total points > 10 billion, size ~ 50 GB
  - 12 threads on a standard Delftblue node
  - sampling method: poisson
  - Compression: BROTLI

netherlands eScience center

PotreeConverter2 MPI



netherlands eScience center

Future work on Delftblue

- Full MPI implementation of PotreeConverter2
  - Trying to improve single node performance by memory mapping the chunk may complicate the MPI implementation
- Solve disk space issue
- Scale Martijn's SFC representation on Delftblue
- Maybe further look into untwine MPI.

netherlands eScience center

Questions and Suggestions?

www.eScienceCenter.nl

n.ahmed@esciencecenter.com

+31 (0)20 460 4770