Traitement de données d'imagerie sur 2 architectures disques différentes au centre de calcul

Marc Betoule, Nicolas Regnault, Pierre Astier

Introduction

We were able to develop our image processing pipeline on a SMP machine at CCIN2P3 (Thank you Fabbio)

Processing

Supernovae detection in HSC image subtraction

Pipeline development

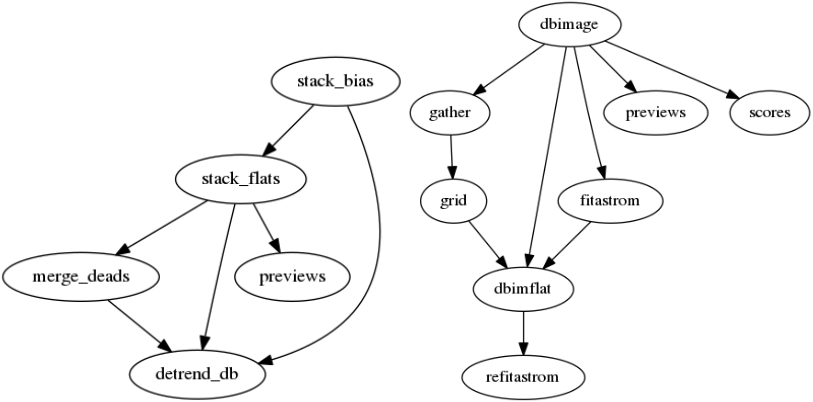
About 20% of the complete data processed on a shared memory computer with large SSD

Large-scale deployment

Same machine but SPS (about 1TB of raw data)

Real case to measure performances of 2 $\neq$ architectures

Our pipeline



Representative of 3 different kind of processing

“Detrending”

(Bias subtraction, flatfielding, . . .)

- Trivial operation on image pixels
- Low number of floating op per byte

Image processing

(Segmentation, photometry . . .)

- Non-trivial operation on image pixels
- moderate number of floating op per byte

Catalogs treatment

(Photometric and astrometric alignment)

- Non-trivial operation on catalog data
- highest number of floating op per byte

The two architectures

Test machine (ccwsge1348)

- 2 Xeon E5-2680
- 14 core / socket (+hyperthreading)
- last level cache 35MB

Local discs

- Raid Perc over PCI-E 12Gb/s (3 SAS HD)
- NVM-Express SSD (3To)

Detached discs

- GPFS (sps)
- passband ?

Implementation details

The pipeline is optimized to avoid unecessary IO

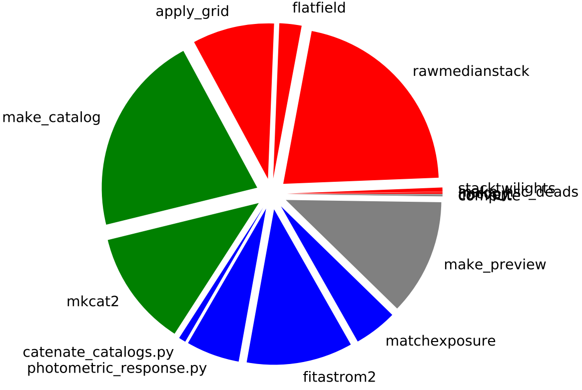
- Whenever a task or a set of task goes several time through the same data
- IO are buffered on a ramdisk
- Only the final result is written to real disks

Measurement principle

- The code itself is not instrumented
- We collect resource usage provided by the kernel
- No penalty/The granularity is rather coarse
- Difficult to get a trace of the actual IO

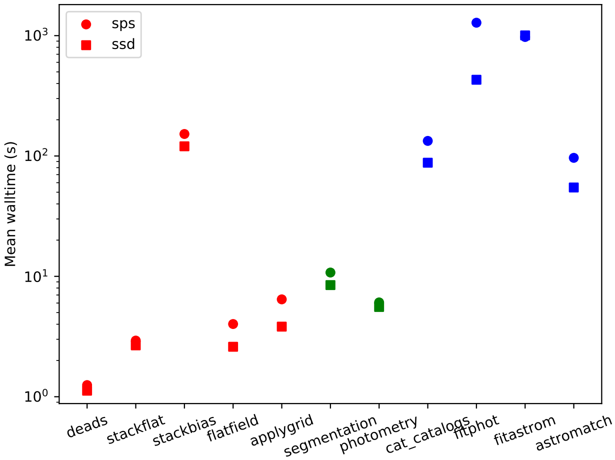
How is time spend

'Detrending' 'Image processing' 'Catalog processing' other

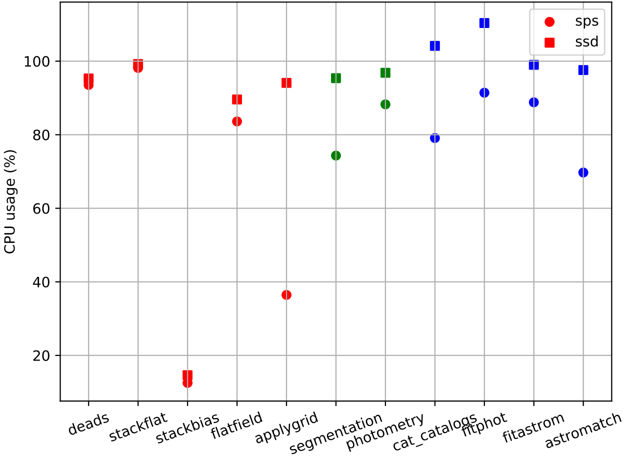


Most of the time is spend in IO-hungry tasks

Mean walltime per task



Is CPU usage efficient ?



What about stacking subtraction ?

- I only have measurement of subtraction efficiency on the local raid of ccwsge1348
- CPU usage average at 87.4%
- Similar to other tasks on the SSD

Conclusion

We measured the throughput of two different architectures in a real case

Raw data to calibrated catalogs:

- Farm: $\sim$ 1MB/s/per core
- ccwsge1348: up to 50% faster

Despite this low number CPUs are easily data-starved

- The SSD / local disks efficiently feed all CPUs (but in the most trivial cases)
- The SPS does not fully feed all CPUs for all tested cases (but for expensive non optimized tasks)

Improvements

- Instrument the pipeline to trace IO (better understanding)
- Redo the measurements using all 28 cores of the computer to see how these excellent results scale to full charge.