Using GPGPU to Increase Accessibility and Efficiency in LHC Computational Systems

February 2015

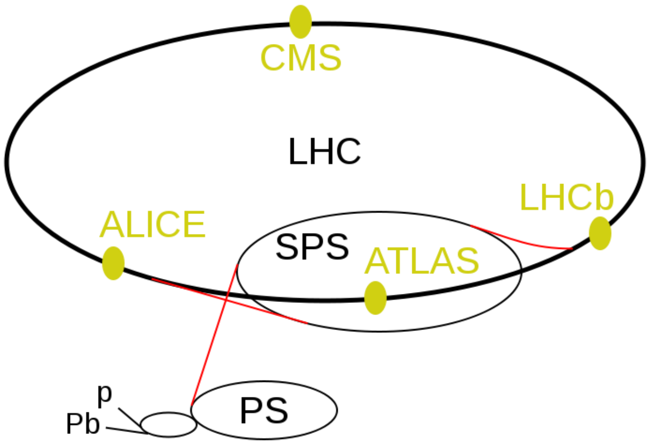
Marc Bawl Sacks

Overview

- LHC (ATLAS) Background
- (ATLAS) Computational Requirements
- Possible solution: parallel computation
- Parallel computation with GPU
- GPGPU and ATLAS
- Conclusion

Large Hadron Collider

- Largest, most powerful particle collider
- 27 km in circumference with four detectors
- ATLAS: A Toroidal LHC ApparatuS

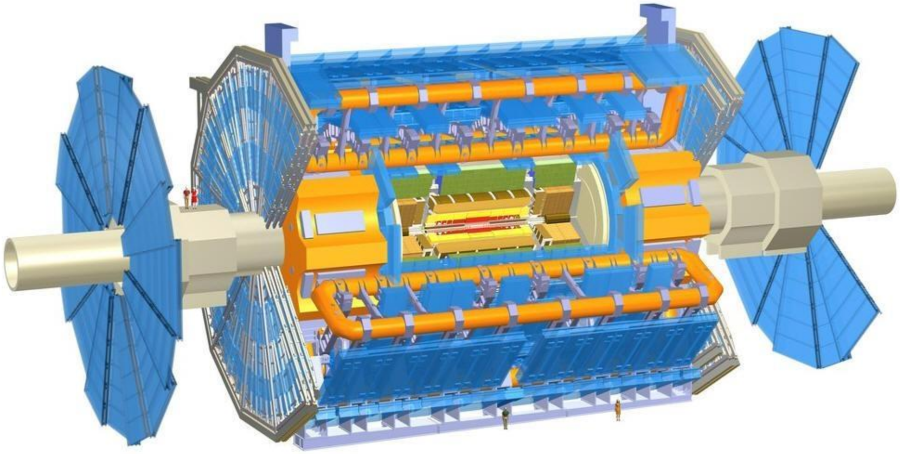


High-energy physics

Low-energy physics

ATLAS Detector

- Composed of several subdetectors
- Tile Calorimeter (TileCal): main hadronic detector
- Measures energy of hadrons, jets, taus, missing tranverse energy
- Bunch crossings at 40 MHz

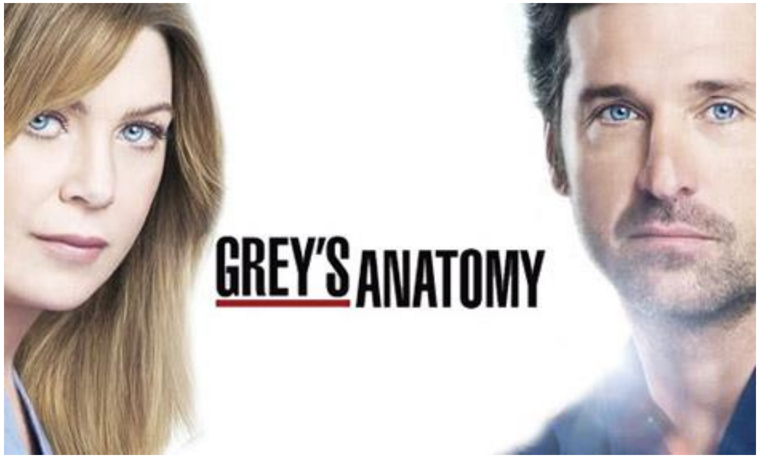


TileCal Detector

- PMT-Scintillator system
- PMT → ADC → L1Calo → ROD (Level 1 trigger)
- ROD → Level 2 trigger → Level 3 trigger
- 40 MHz → ~200 Hz

TileCal Upgrade

- Currently first level trigger deals with 205 Gbps
- Upgrade will deal with 41 Tbps; Google deals with ~1.85 Tbps
- ~3500 episodes of Grey's a second
- Obvious need for high-throughput computing: parallel computing



Parallel Computing

- Uses divide and conquer versus brute speed
- Becoming evermore ubiquitous: multicore but even ILP in singlecore
- Example:

```
for i=1, i<=100, i++
{ array[i]=i*i }
```

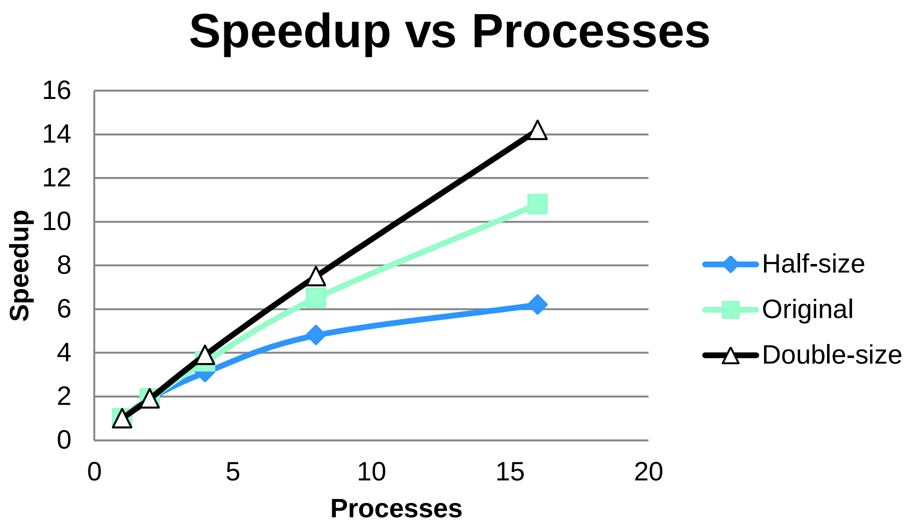
One 10 Hz processor: 10 seconds

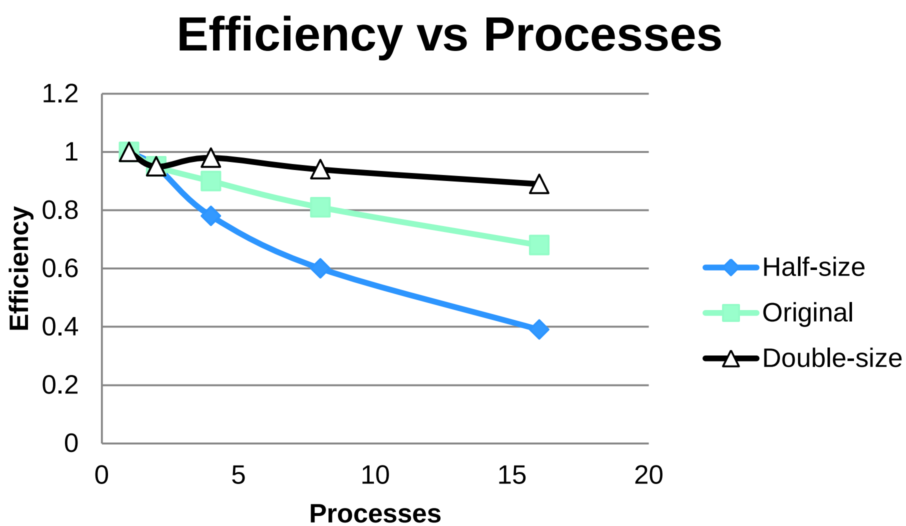
Ten 2 Hz processor: 5 seconds

Speed Metrics

- Serial, parallel runtime: $T_{serial}, T_{parallel}$
- Linear speedup: $T_{parallel} = \frac{T_{serial}}{p}$, $p =$ no. of processes
- More realistically $T_{parallel} = \frac{T_{serial}}{p} + T_{overhead}$
- Efficiency $E = S/p$

Speed Metrics





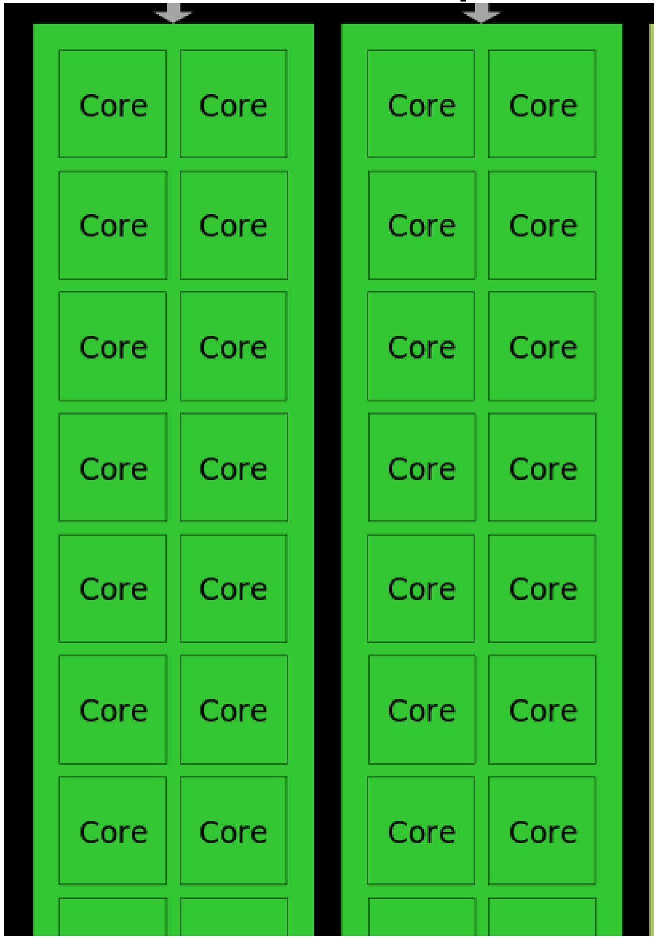

```
for i=1, i<=10,
i++
{ array[i]=i*i }
```

One 10 Hz processor: 1 second

Ten 2 Hz processors: 2 seconds

Parallel Computing with GPU

- GPU architecture lends itself to parallel computing.

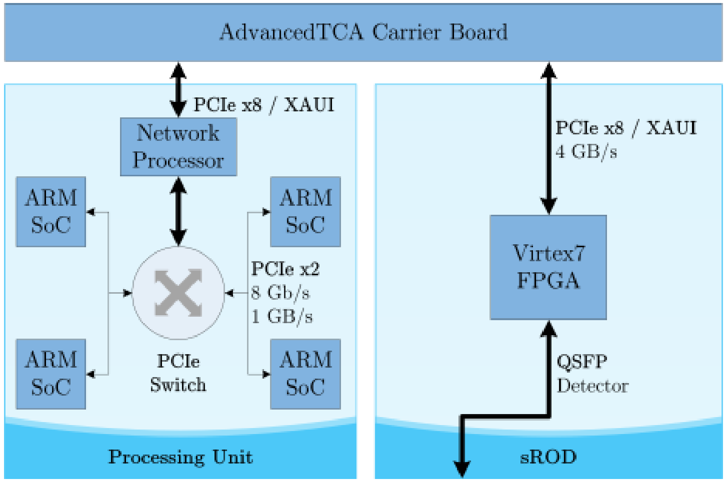


What Problems Fit GPU?

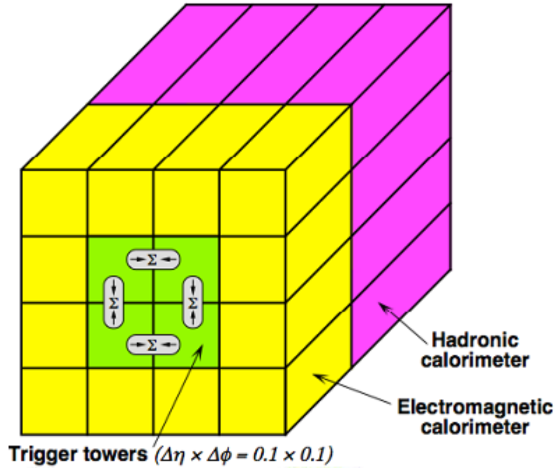
- Embarrassingly parallel
  - Image processing, Mandlebrot, Monte Carlo
- Serial programs with parallel parts, serial to parallel
- TilCal and GPGPU
- Higher level triggering and simulations

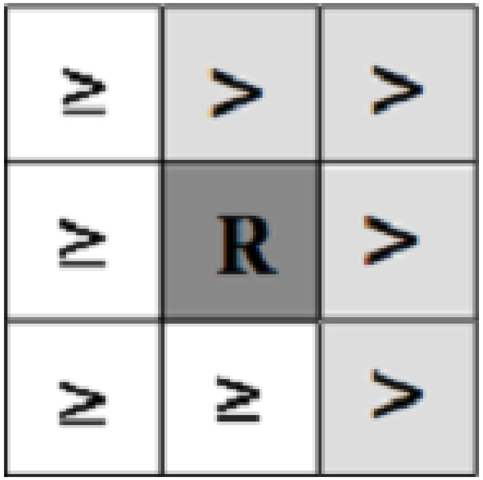
GPU and the TileCal PU

- 224 trigger hardware components are read out by 20 RODs
- RODs are being upgraded to sRODs



Trigger Logic as Image Processing





GPU Accessibility

- GPUs are relatively cheap and easily acquirable
- High-level and low-level C-like languages
- Plethora of literature
- Growing every day

Conclusions

- In 2022 TileCal will produce 41 Tbps
- Need a way to deal with this online
- Parallel computing could assist
- GPGPU is a promising tool with the PU and more

Thank you!

Please direct all questions and comments to The Broadcasting Complaints Commission of South Africa:

PO Box 412365, CRAIGHALL, 2024