Babar and SuperB application runtime behaviour

Vincenzo Ciaschini

SuperB computing workshop

4-7/7/2011

What is this?

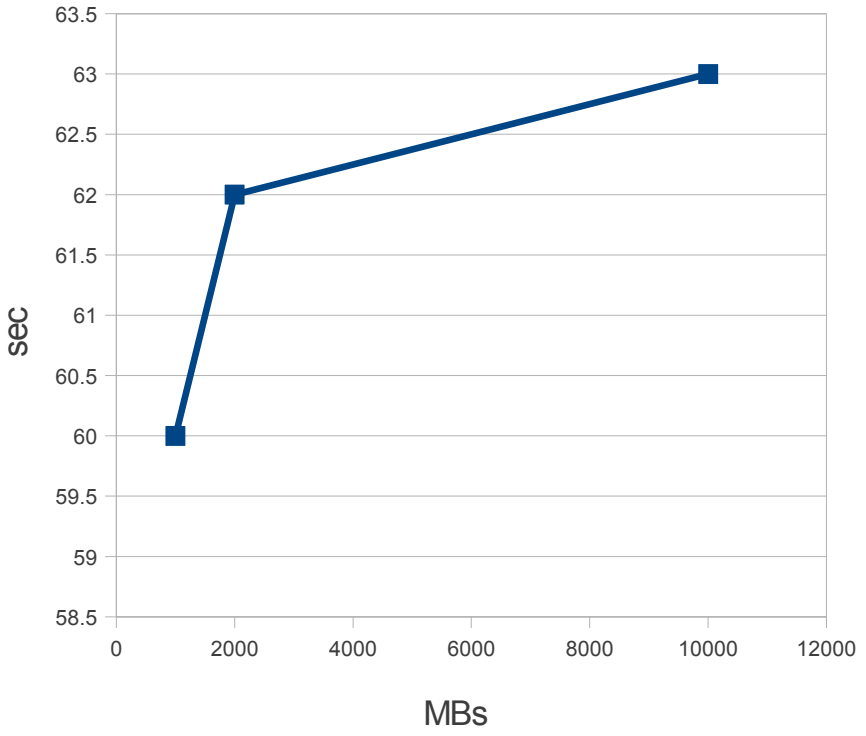
- With help from Armando and Marco, at CNAF we run some typical SuperB and Babar applications to see what would be the constraints when scaling up to many cores.
- These are the results

Metodology

- For Babar tools (SkimMini)
  - Run on official Babar machine, with official binaries (32bit only)
    - SL 5.3, 1 core, 2,2 Ghz, 2 GB
- For SuperB tools (FastSim and PacUserApp, and contention)
  - Run on official SuperB machine, with official binaries (64bit)
    - SL 5.3, 8 core 1,8 Ghz, 8 GB
- For all
  - Used igprof for profiling
  - Used iostat for I/O data
  - Used time for timing

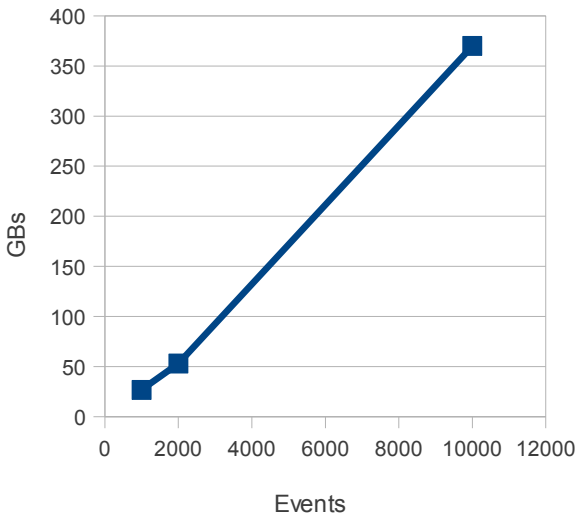
SkimMini

Time and Memory

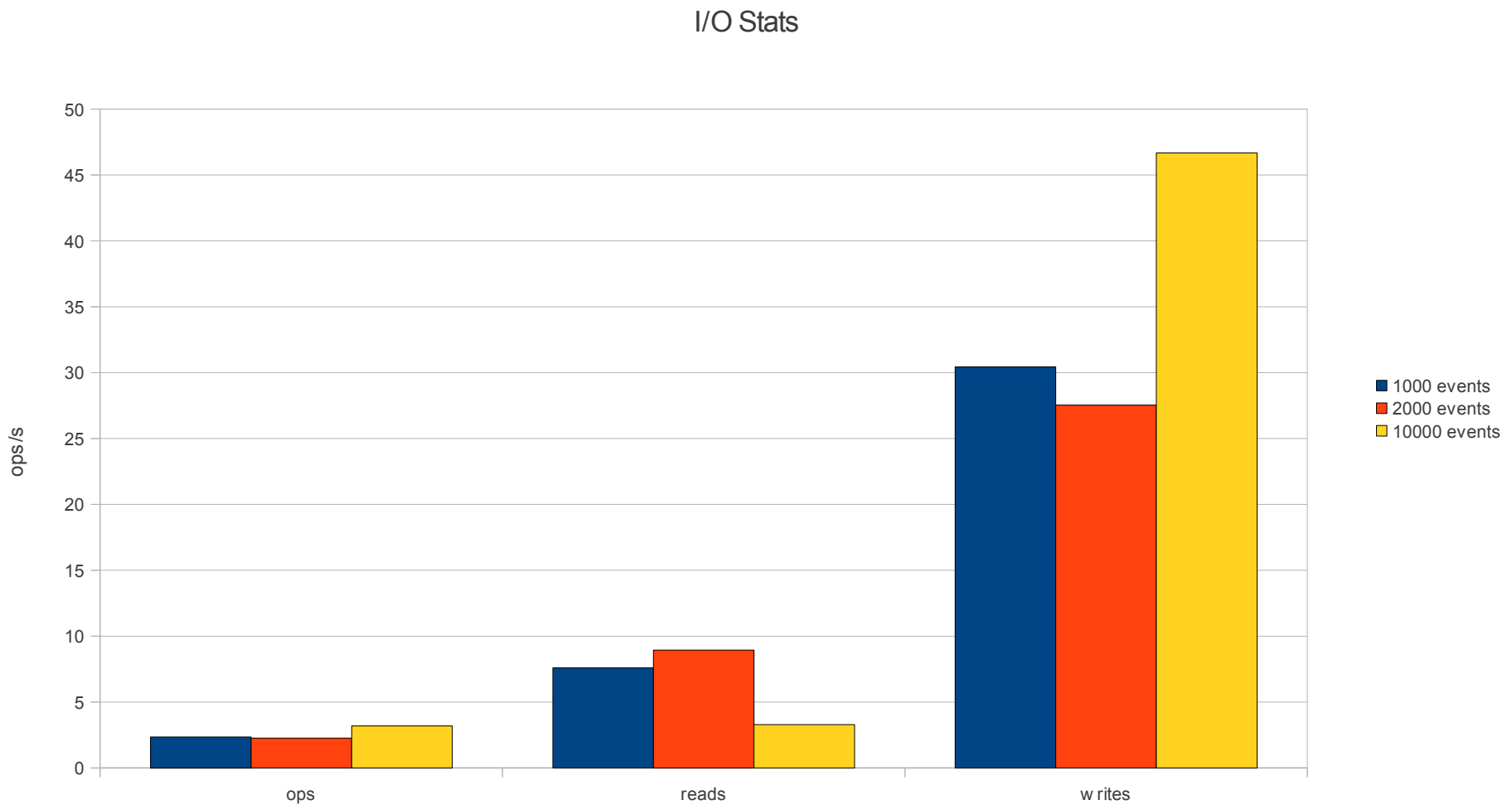


Memory usage analysis

- Memory usage increase seems slow
  - Mostly dominated by CINT (27 MB)
  - Should be interesting:
    - Calls to CLHEP::HepMatrixAllocBase::myMalloc()
      - Single greatest increasing leaks



I/O

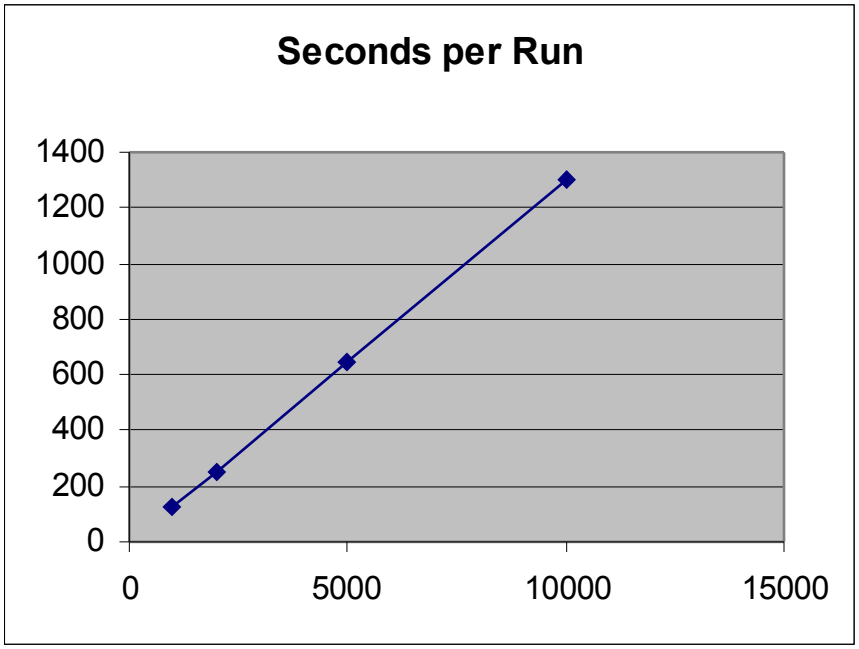

I/O Stats
ops/s
50
45
40
35
30
25
20
15
10
5
0
ops
reads
w rites
1000 events
2000 events
10000 events

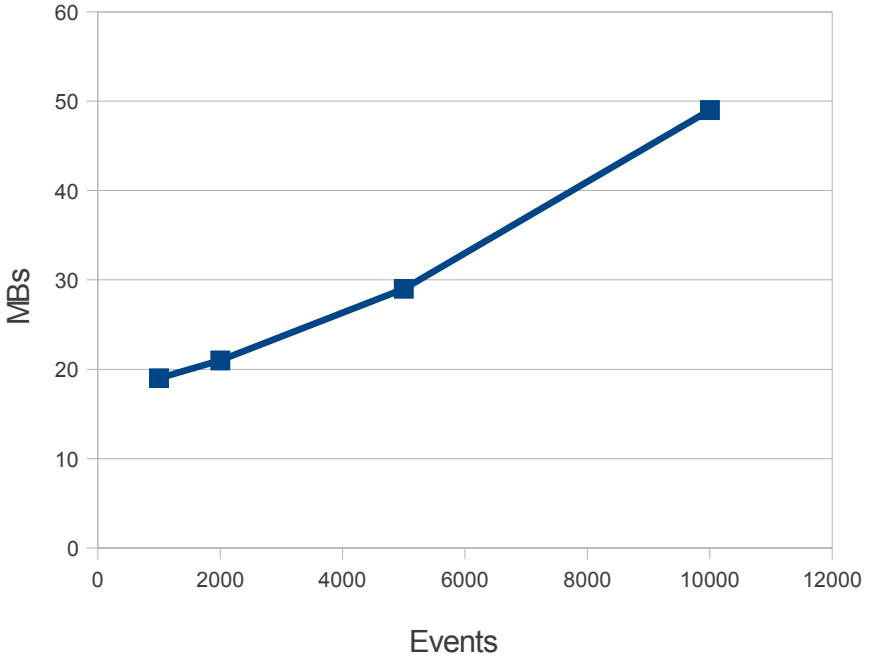
I/O Note:

- The first minute of activity was cut out of the data.
  - Initial ramp up caused read I/O on the order of 10/20 times that of the following minutes

FastSim

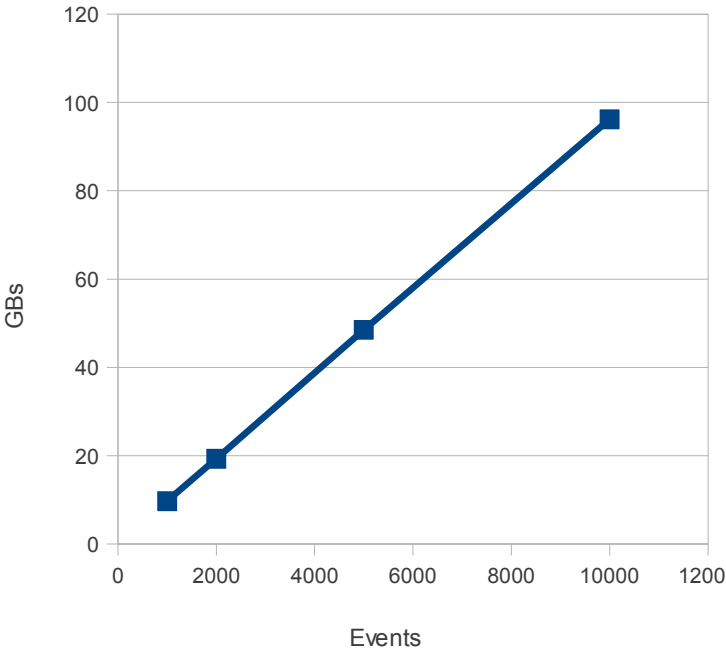
Time and Memory



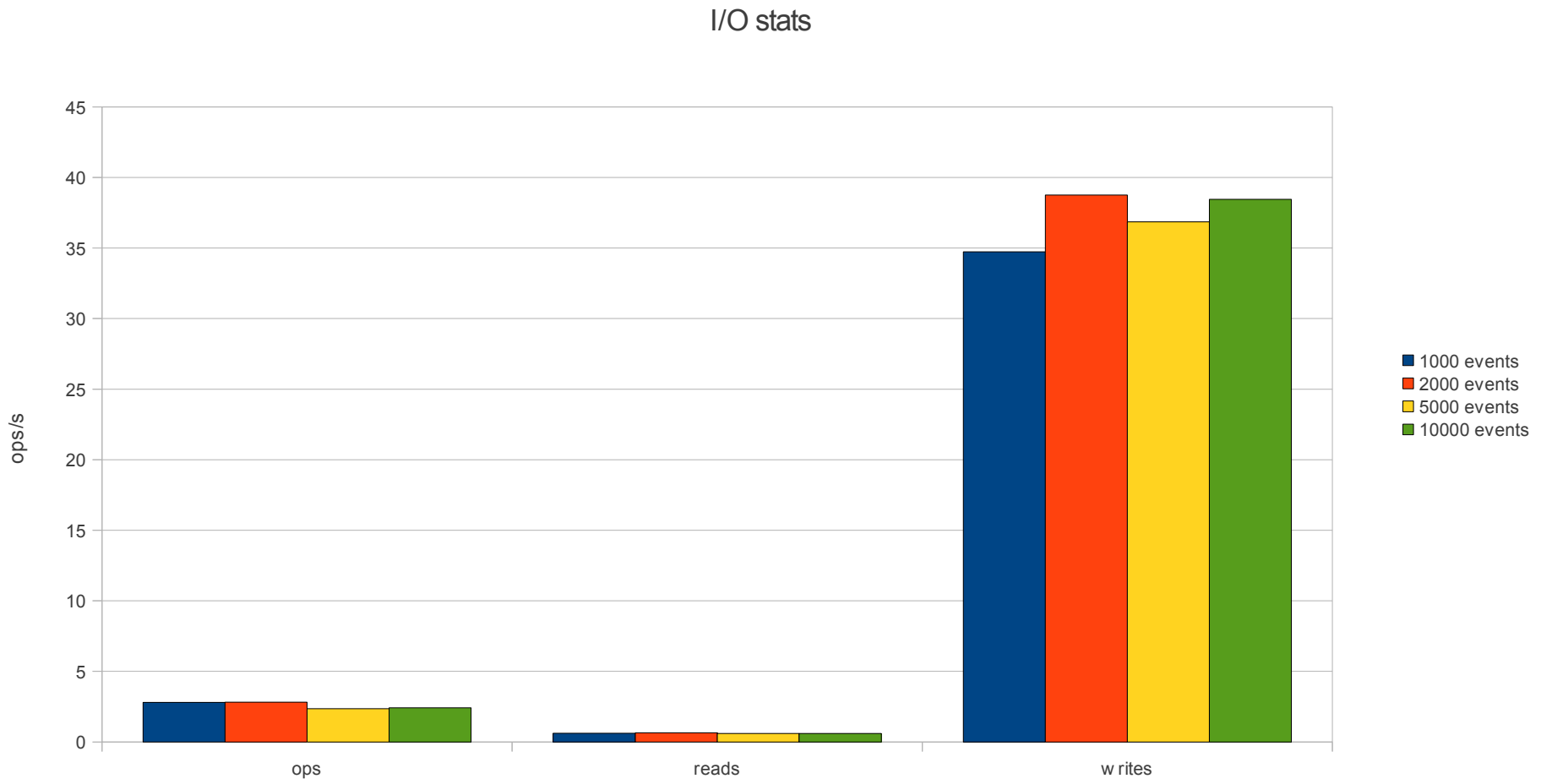


Memory Usage Analysis

- Memory usage increases quite fast (around 2Kb per event)
  - Mostly linear
  - Most responsible functions:
    - PacTrkHitMeas::createHots()
    - PacHitOnTrk::PacHitOnTrk()



I/O

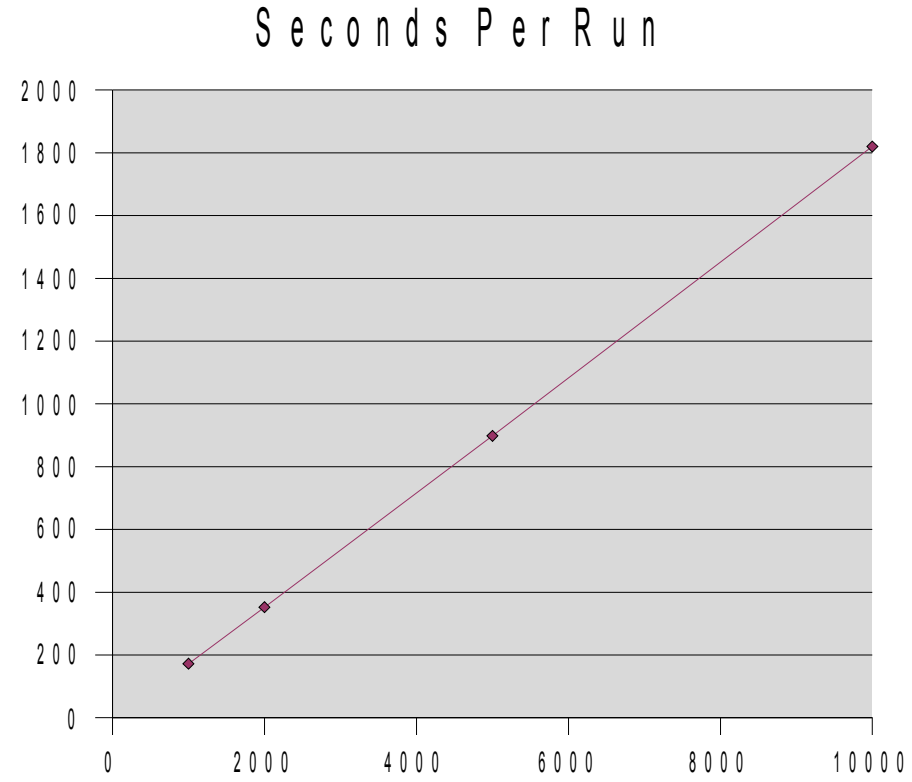

I/O stats
ops/s
45
40
35
30
25
20
15
10
5
0
ops
reads
w rites
1000 events
2000 events
5000 events
10000 events

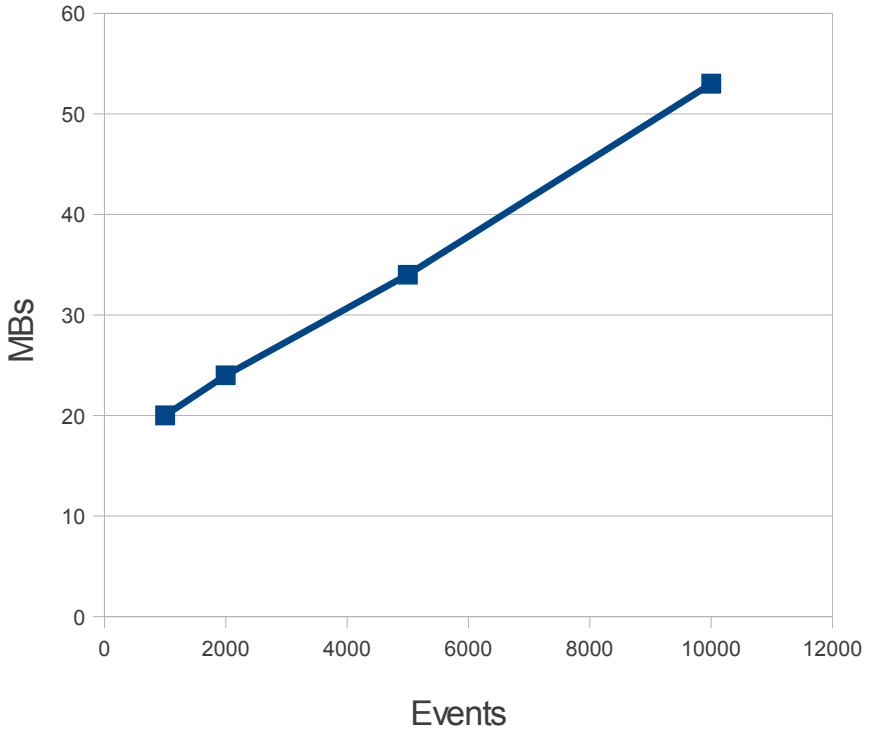
I/O Note:

- The first minute of activity was cut out of the data.
  - Initial ramp up caused read I/O on the order of 10/20 times that of the following minutes

PacUser

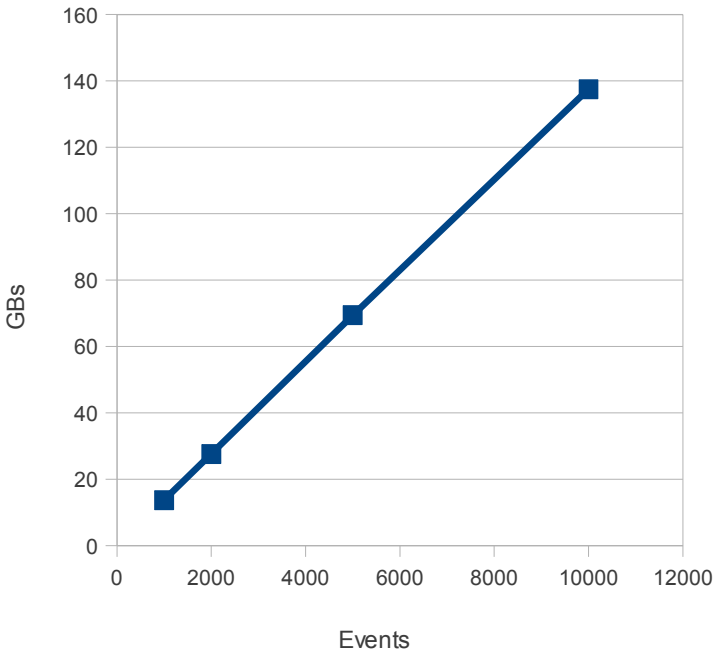
Time and Memory



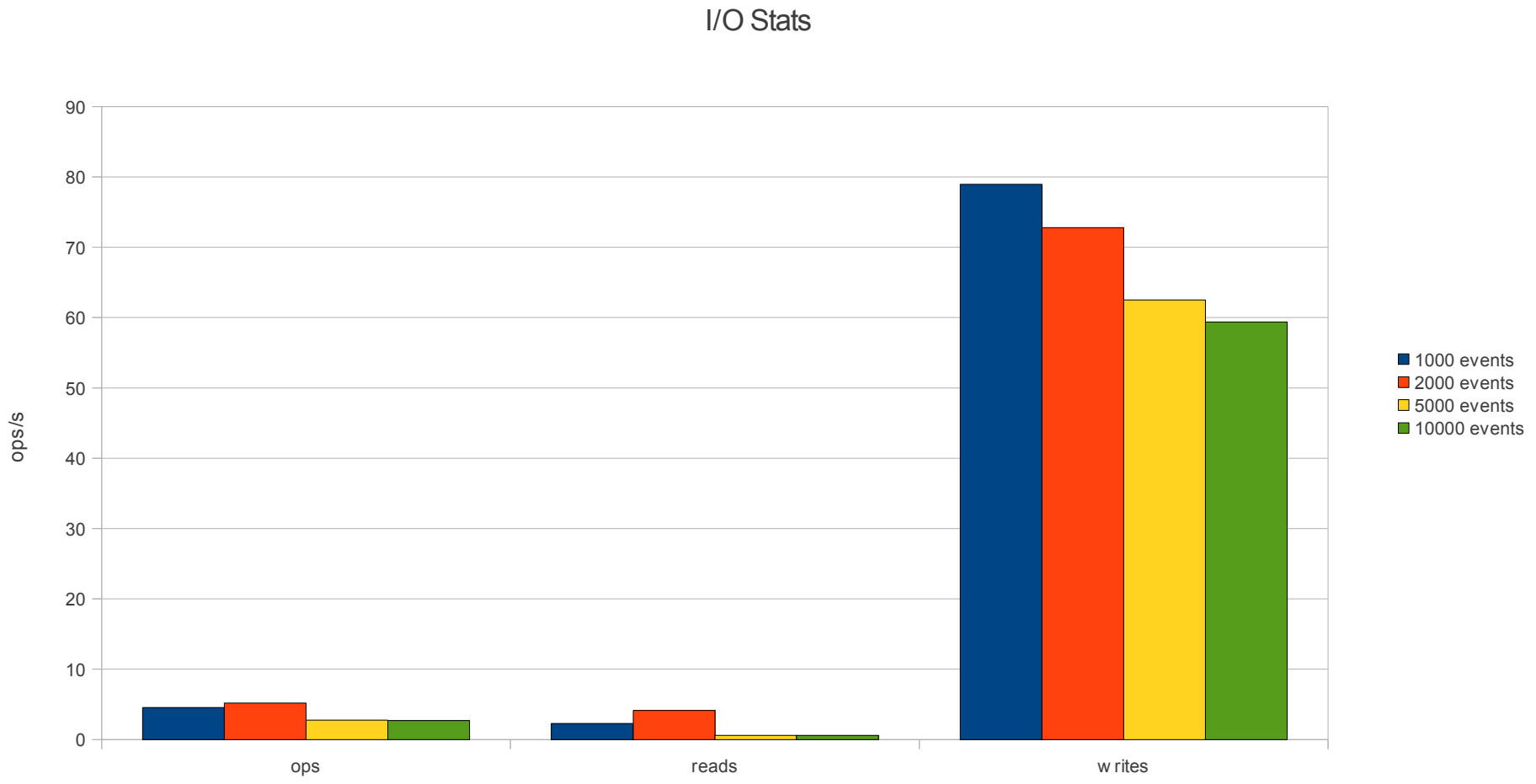


Memory usage Analysis

- Memory usage increases quite fast
  - Mostly linear
  - Most responsible functions:
    - PacTrkHitMeas::createHots()
    - PacHitOnTrk::PacHitOnTrk()



I/O

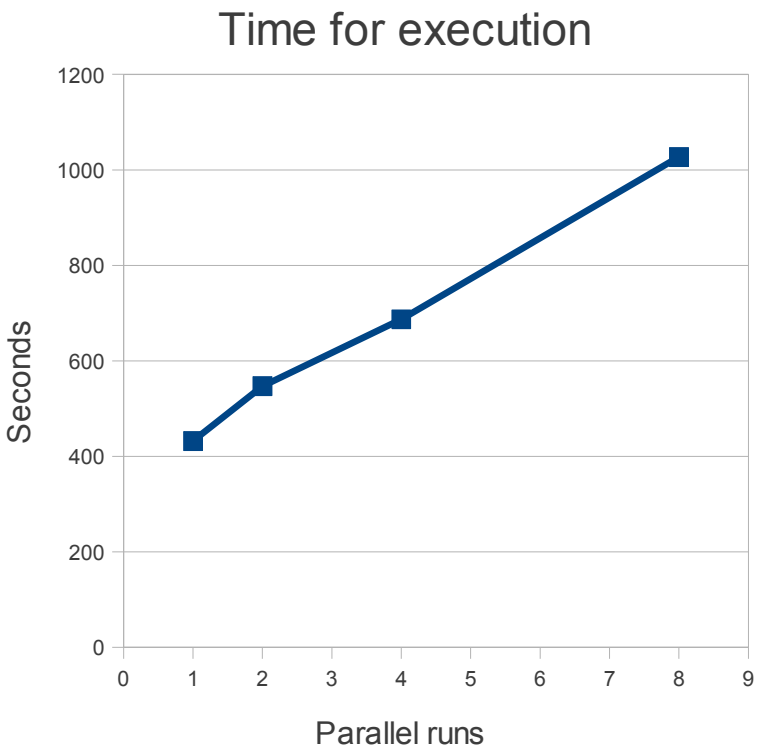

I/O Stats
ops/s
90
80
70
60
50
40
30
20
10
0
ops
reads
w rites
1000 events
2000 events
5000 events
10000 events

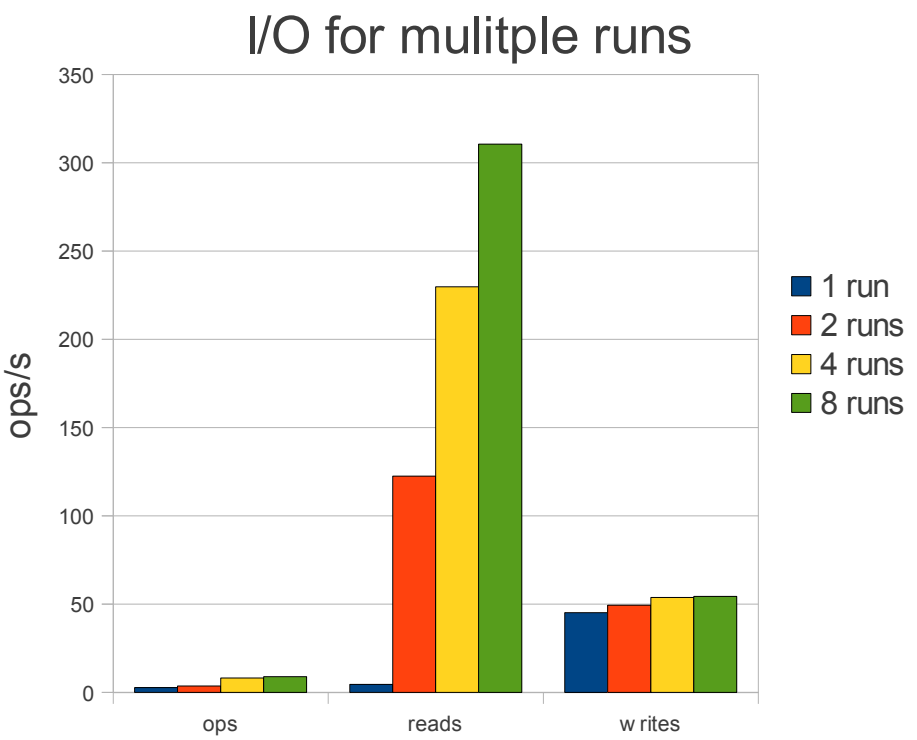
Contention

How?

- On a 8 core machine
  - Run 8 copies of SkimMini in background at the same time, with 1000 events
  - Run 8 copies of PacUserApp in background at the same time, with 1000 events

SkimMini

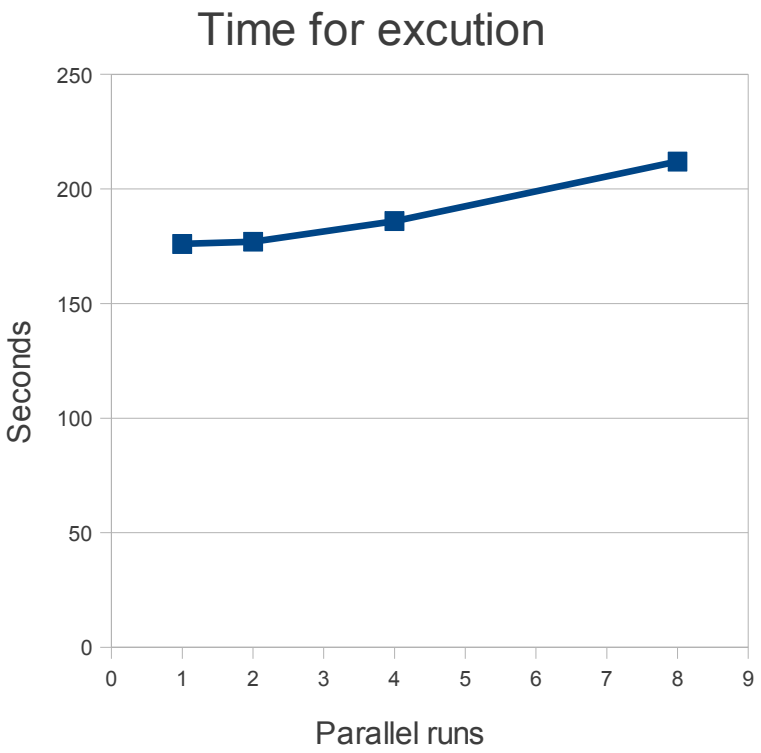

Time for execution
Seconds
Parallel runs
0
200
400
600
800
1000
1200
0
1
2
3
4
5
6
7
8
9

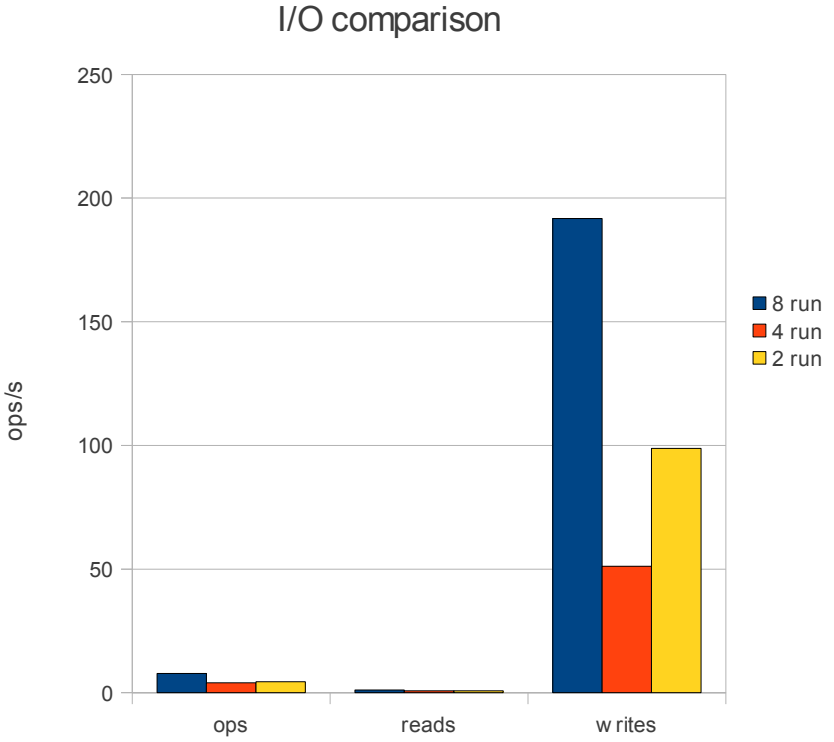

I/O for mulitple runs
ops/s
0
50
100
150
200
250
300
350
ops
reads
w rites
1 run
2 runs
4 runs
8 runs

SkimMini Comparisons

- Time: > 1000 sec (> x2 times)
- CPU: < 5% for first 75%, > 99% for the rest
- Not much reads, however (after the first 3/4 minutes), writes are constant
  - Memory contention ?

PacUserApp


Time for excution
Seconds
250
200
150
100
50
0
0
1
2
3
4
5
6
7
8
9
Parallel runs


I/O comparison
ops/s
250
200
150
100
50
0
ops
reads
w rites
8 run
4 run
2 run

PacUserApp Comparisons

- Time: 212 seconds (roughly 25% increase)
- CPU: > 99% (against 14%)
- I/O: increases and decreases

Thanks

- Many thanks to Armando Fella and Marco Corvo for their help.
  - But I lay claim to all the errors!