Scalability Problems In Shared Memory

José Alves, Rui Brito

Universidade do Minho

Braga, March 2013

Index

The OpenFoam Computational Fluid Dynamics (CFD) software package

- Highly Modular
- By scientists, for scientists
- Fluid dynamics problems, involving chemical reactions, turbulence and heat transfer
- Has many other applications

Memory Bandwidth

$Mem\_bandwidth = ((PAPI\_LLC\_TCM) \times 64 \times 1 \times 10{-}9)/exec\_time$
$measured\_mem\_bandwidth = (3826100 \times 64 \times 1 \times 10^{-9})/0.942194 = 0.2599 GB/s$
$System\_memory\_avail\_bandwidth = 14.8473 GB/s$

Instructions per cycle

$Total\ Instructions = 211932871\ Number\ of\ cycles = 1548$
$Instruction\ per\ cycle = 136907$

As seen in the December presentation, conv-diff has some locality problems (has it can also be seen by the high number of L1 misses (14937146))

The OpenFoam Computational Fluid Dynamics (CFD) software package

- Highly Modular
- By scientists, for scientists
- Fluid dynamics problems, involving chemical reactions, turbulence and heat transfer
- Has many other applications

Memory Bandwidth

$Mem\_bandwidth = ((PAPI\_LLC\_TCM) \times 64 \times 1 \times 10{-}9)/exec\_time$
$measured\_mem\_bandwidth = (3826100 \times 64 \times 1 \times 10^{-9})/0.942194 = 0.2599 GB/s$
$System\_memory\_avail\_bandwidth = 14.8473 GB/s$

Instructions per cycle

$Total\ Instructions = 211932871$ $Number\ of\ cycles = 1548$
$Instruction\ per\ cycle = 136907$

As seen in the December presentation, conv-diff has some locality problems (has it can also be seen by the high number of L1 misses (14937146)).

Beat

Vender peixe, basicamente

1. Com alta modularidade entende-se que é composto por várias bibliotecas, e cada função da biblioteca pode ser aplicada a uma grande variedade de tipos de dados. E ser resolvido com solvers e esquemas diferentes

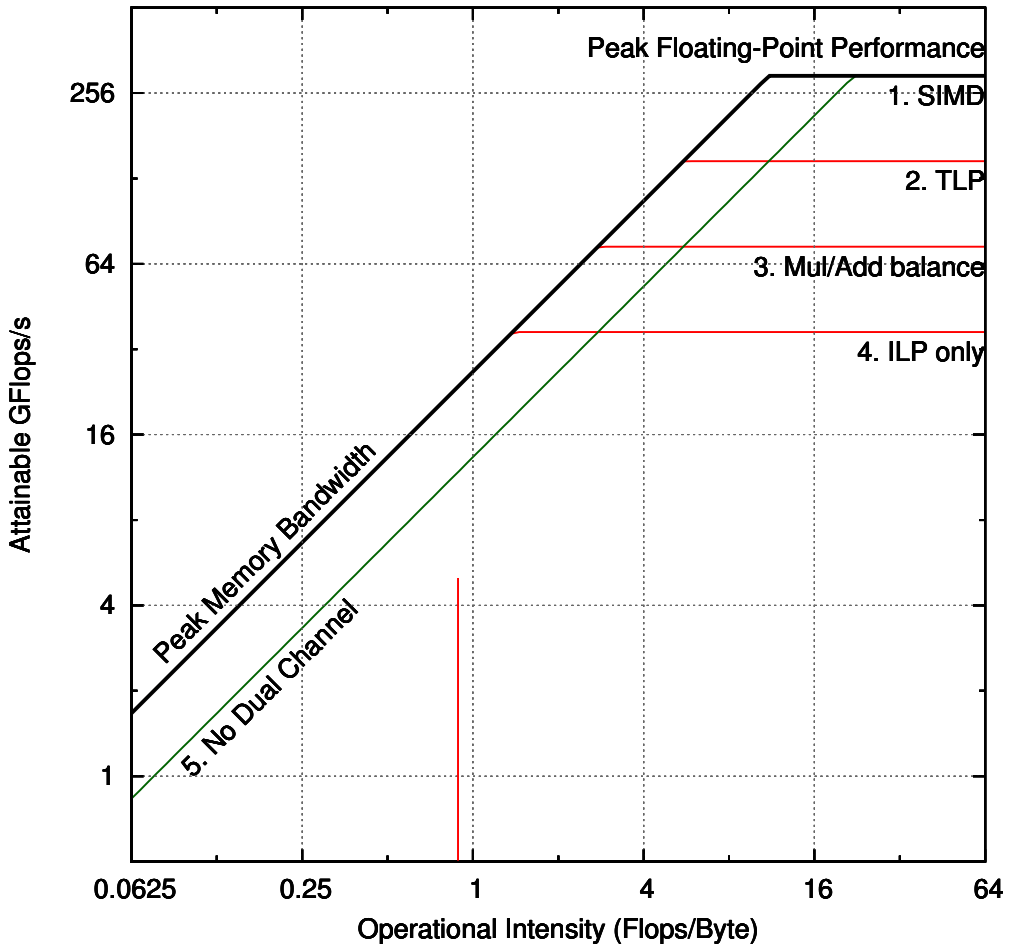

Figure : Roofline for rMBP and conv_diff

Task Granularity

In conv_diff:

- Only two parallel pieces of code run in parallel
- Large chunks of code
- Thread creation overhead is thus minimized

conv_diff is unsuitable for this sort of optimization:

Excessive task synchronization

- Reduction can't be used because values are updated in an array of pointers
- Synchronization must be forced on attributions

Loads Per Task

- Slight improvement using dynamic and guided scheduling
- Workload distribution also isn't the biggest problem

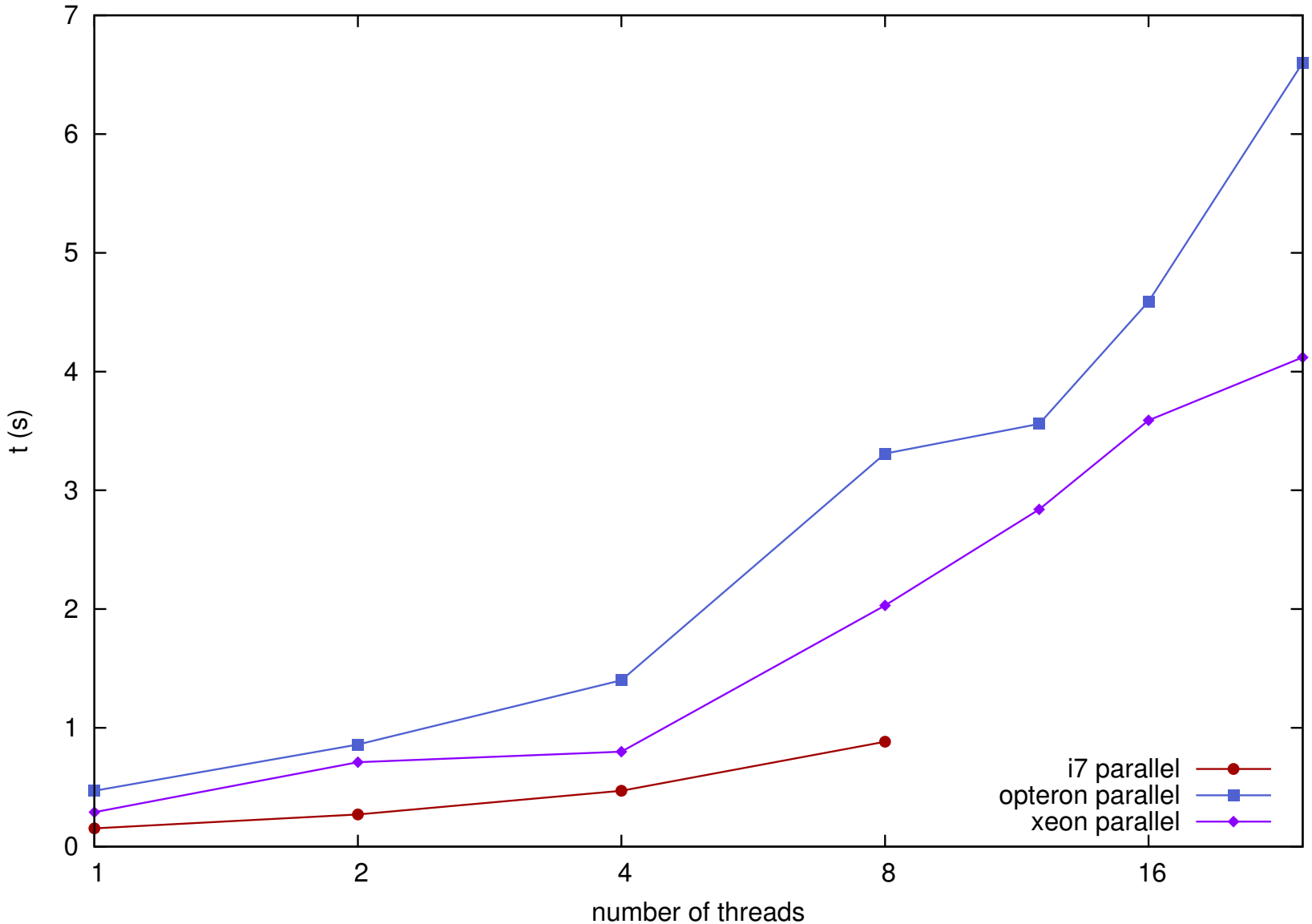

Figure : Scalability of the parallel region (original implementation)

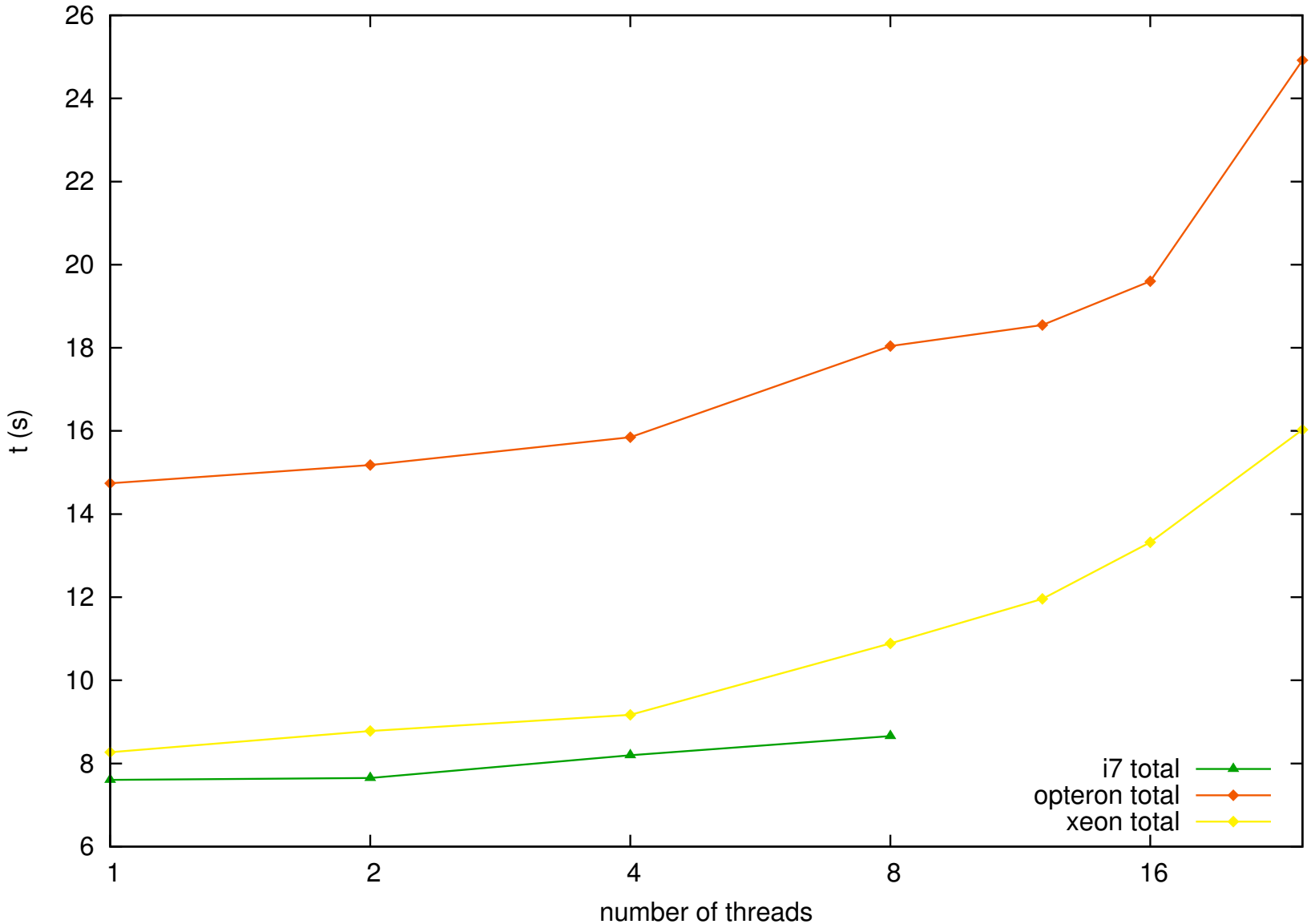

Figure : Total execution time (original implementation)

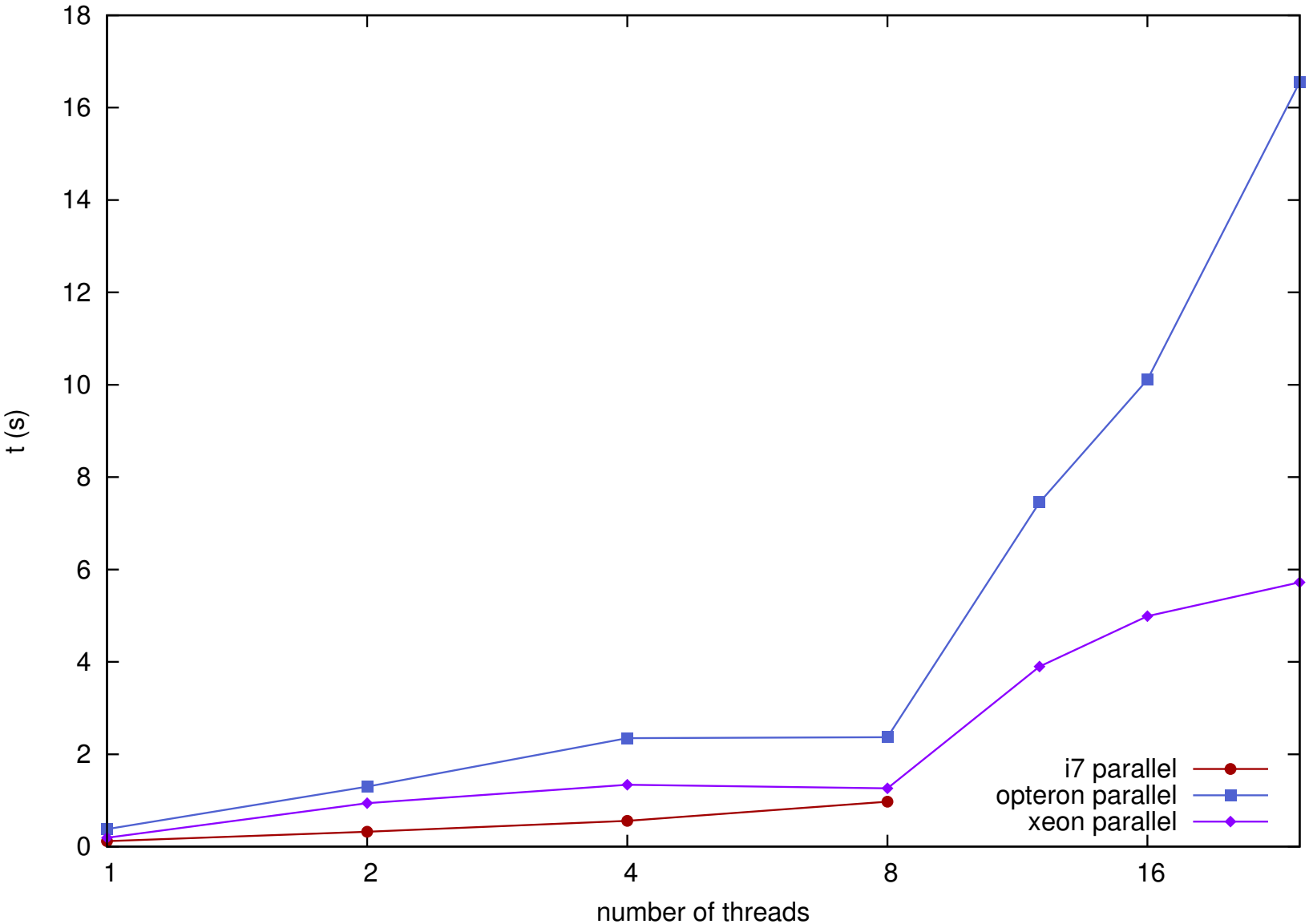

Figure : Scalability of the parallel region (without atomic directive)

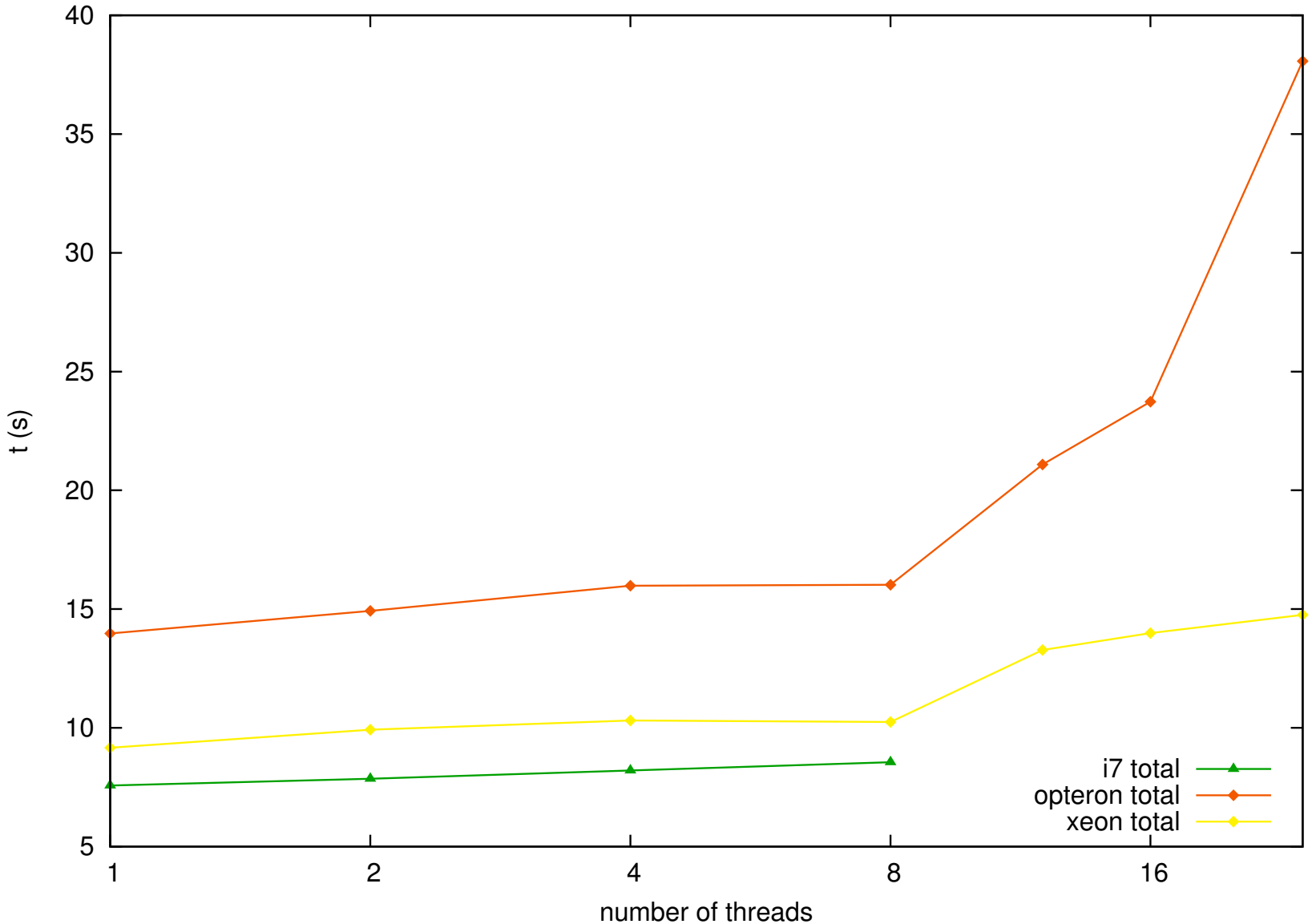

Figure : Total execution time (without atomic directive)

Conclusion

- Bad data locality hinders the entire application performance
- Implementation either AoS or SoA is expected to improve performance dramatically
- High chance that locality problem hides other problems
- Maybe what is not a problem now will prove to be so in the near future

Scalability Problems In Shared Memory

José Alves, Rui Brito

Universidade do Minho

Braga, March 2013

– ? –