Effective Parallel Code Generation Through Auto-Sequentialisation

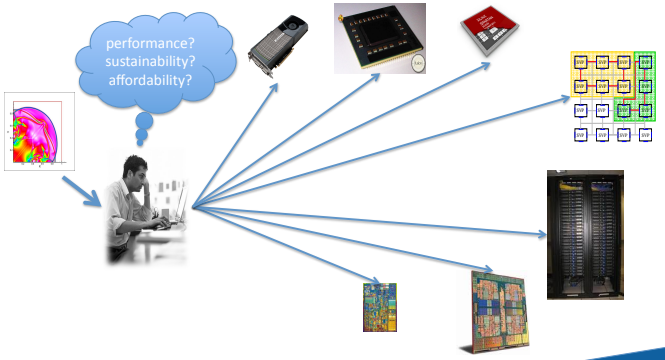
IFIP 2.11 Meeting
27. June. 2012, Halmstad

Stephan Herhut, Carl Joslin, Jing Guo, Dan Rolls,
Clemens Grelck, Raphael Poss, Mike Lankamp,
Chris Jesshope, Sven-Bodo Scholz

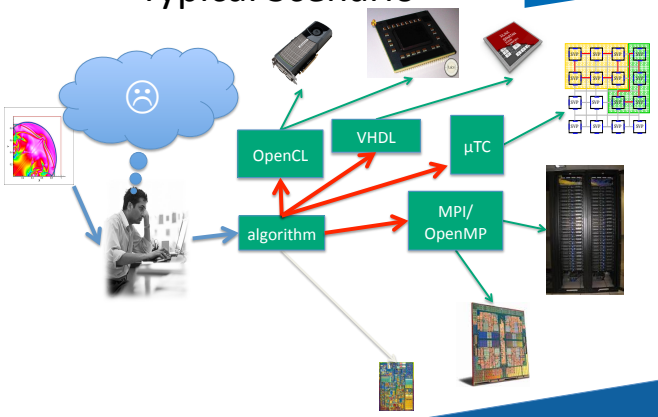
ADVANCE

apple core

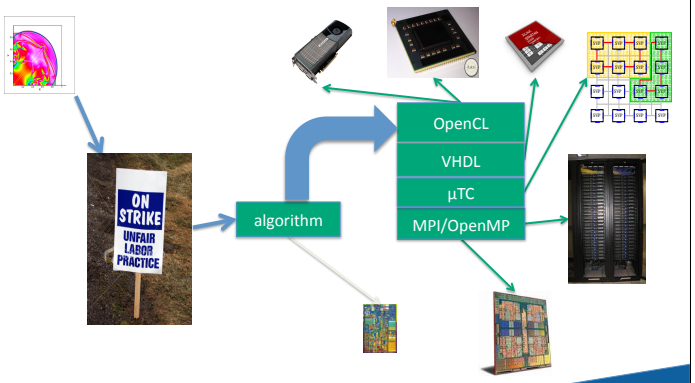
The Multicore Challenge



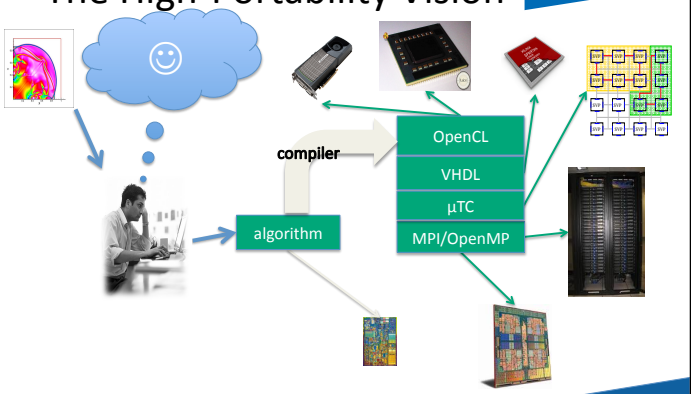
Typical Scenario



Tomorrow's Scenario



The High-Portability Vision



The Parallelism Myth

sequential program parallel code

5

Sequentialisation Challenge

concurrent program

parallel code

6

The Parallelism Challenge

concurrent program

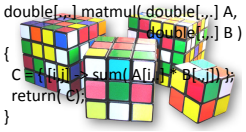
we start here!

sequential program parallel code

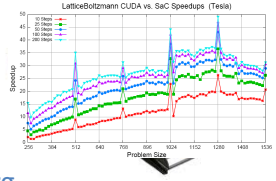
7

What is SaC ?

```
double[.,.] matmul( double[.,.] A,
                    double[.,.] B )
{
  C = { [i,j] -> sum( A[i,.] * B[.,j] ) };
  return( C);
}
```



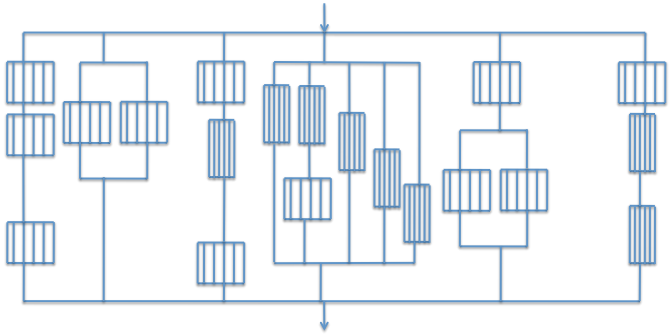
```
a = [1,2,3,4];
b = modarray( a, [0], 42);
stdout = print( a, stdout);
stdout = print( b, stdout);
```



www.sac-home.org

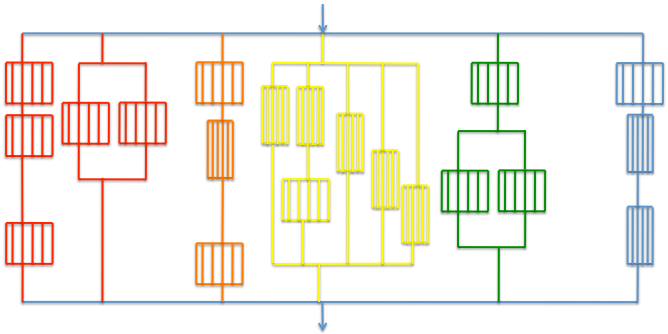
8

Sequentialisation Challenge



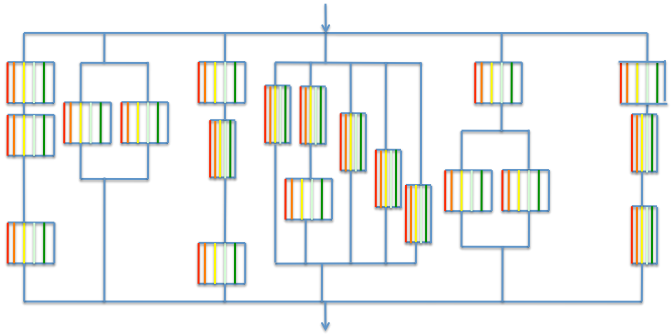
10

Sequentialisation for SMP



11

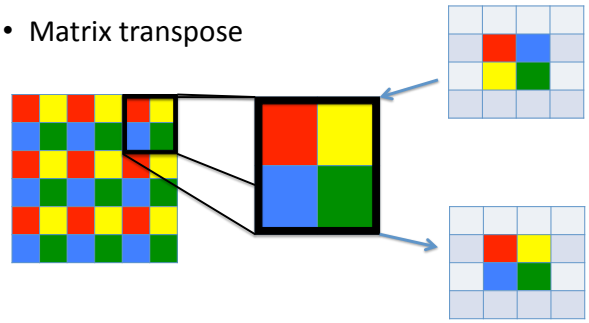
Sequentialisation for GPUs



12

The Memory Abyss

- Matrix transpose

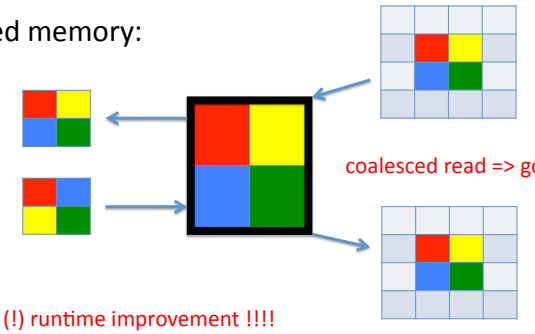


un-coalesced read => bad!!

13

Shared Memory To The Rescue!

shared memory:

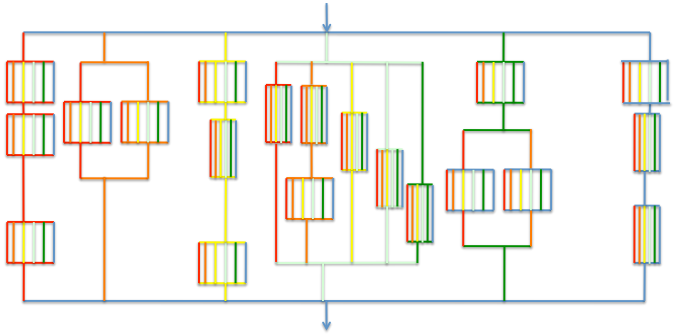


coalesced read => good!!

6-fold (!) runtime improvement !!!!
knowledge about hw-sequentialisation is key!

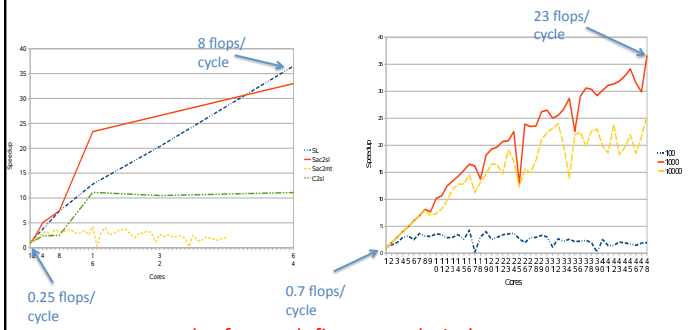
14

Microgrid == Nivana ?!



15

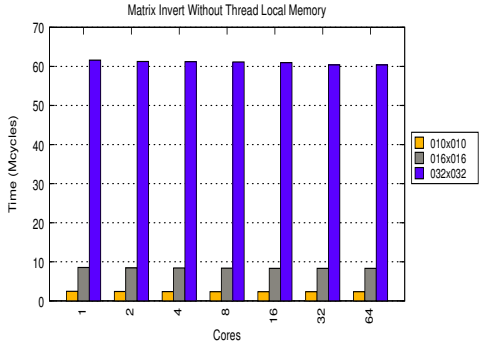
Stencil Computations



scales for much finer granularity!
much better energy efficiency!

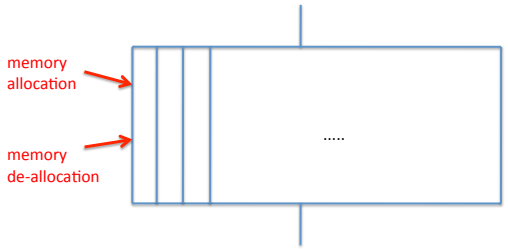
16

The Memory Strikes Back I



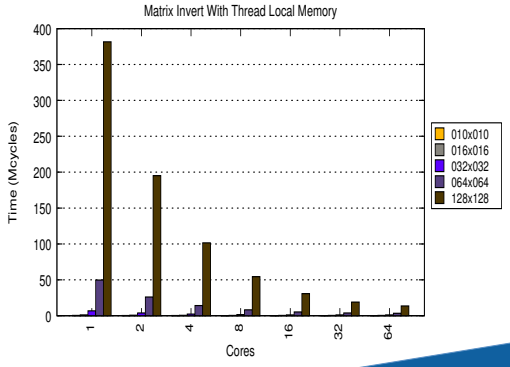
17

What is happening?



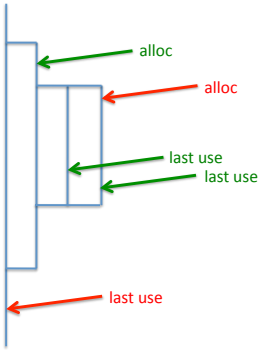
18

HW to the Rescue: Thread-Local Memory



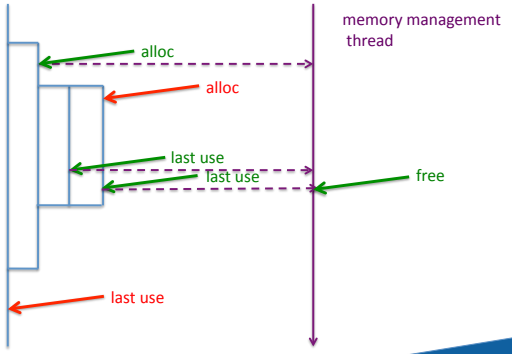
19

Moving to Task Parallelism



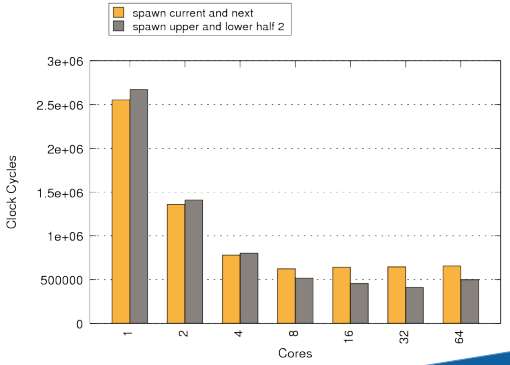
20

Asynchronous Reference Counting



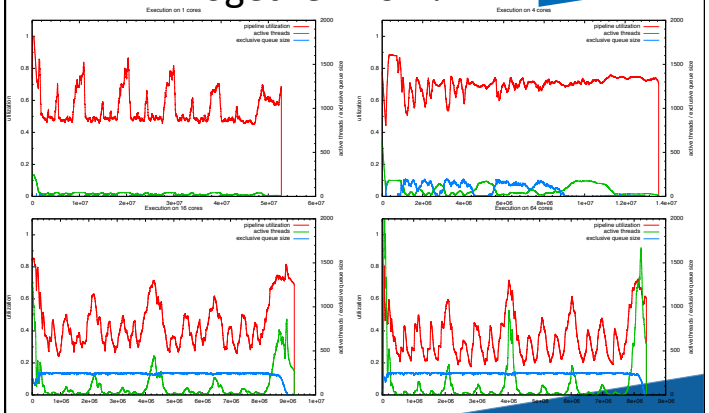
21

Naive fib(12) !



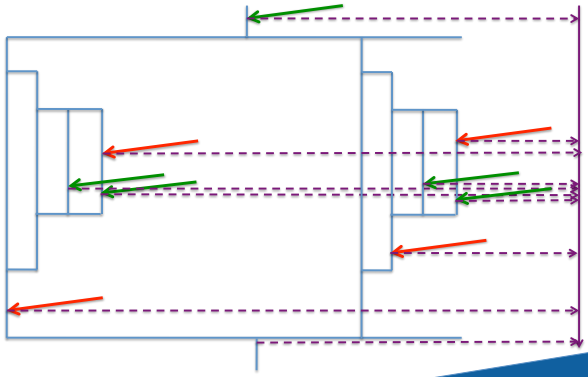
22

All Together Now!



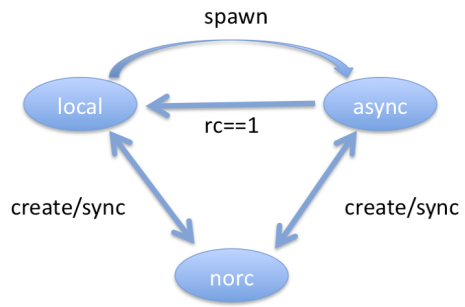
23

Memory, Again!



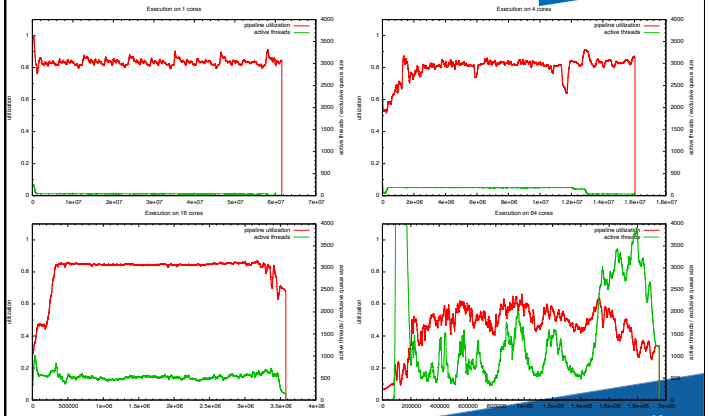
24

Multi-Modal RC:



25

new runtimes:



26

Memory Management and Many Cores

today:

tomorrow:

27

Conclusions

- There is no one-fits-all parallelisation!
- The more parallel it gets the more attention memory organisation requires!
- Sequentialisation and memory organisation go hand in hand!

28