

Smart OpenMP

Alejandro Duran

*CEPBA-IBM Research Institute
Computer Architecture Department
Universitat Politecnica de Catalunya*

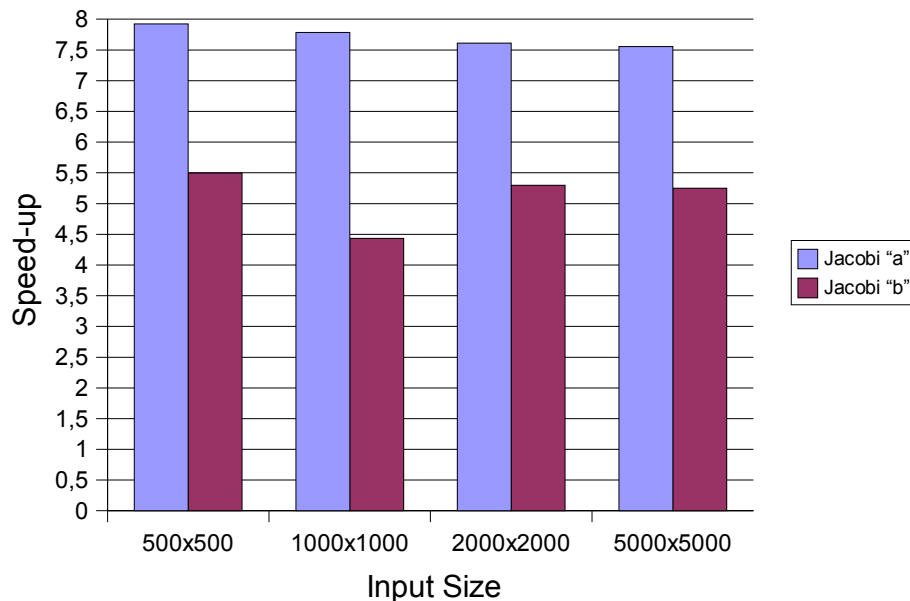
OpenMP forecast

- New architectures will increase the parallelism demand
 - Outside scientific domains
- Not enough offer to cope with the demand
 - Programmers barely know what parallelism is
 - They could learn...
- They'll do the easier & quicker
 - no fancy stuff just defaults

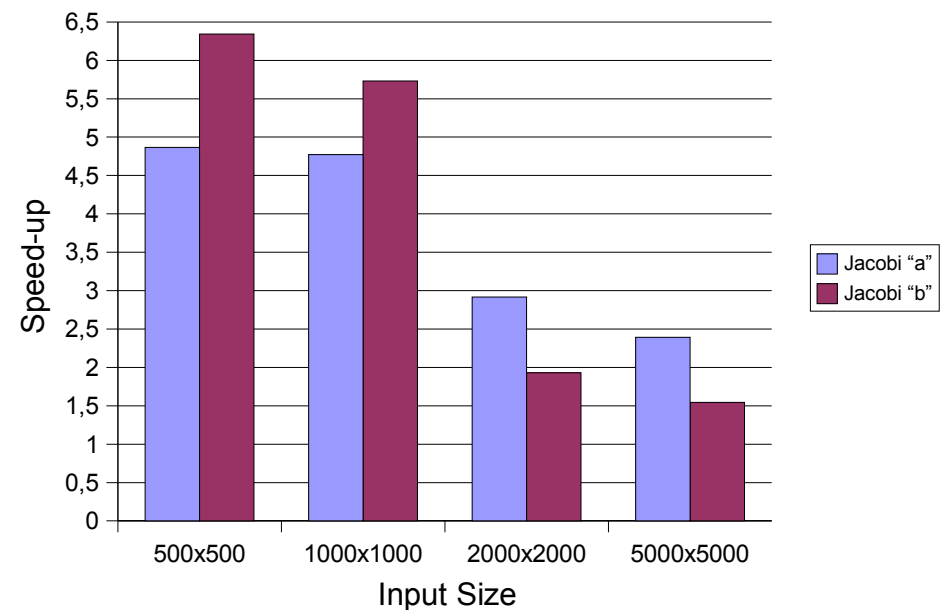
Portability vs Performance

- Data set
- Architecture
- My little jacobi history

CMP (8 threads)



SMT (8 threads)



OpenMP 3.0

- Decouple exploitation from semantics
- Allow plenty of room for dynamic optimizations by default
 - Smart exploitation
 - Smart schedules
 - Smart blocking
 - Smart data layout
 - ...
- Bottom line: Make *defaults* performance-wise!!!

OpenMP 3.0

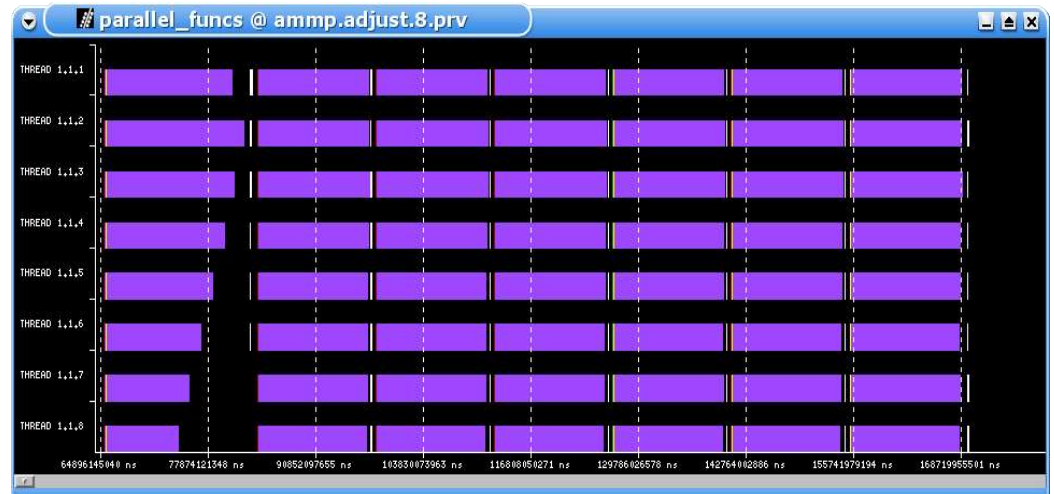
- Already “supported” in some cases
 - schedule by default is implementation dependent
- Expand the cases
 - parallel definition
 - number of threads
 - OMP_DYNAMIC should be default
 - ...
- Encourage the runtime/compiler work
 - reword “implementation dependent” to “application-architecture dependent”

Smart Parallel Exploitation

- **Compiler/Runtime decides best exploitation**
 - How many threads to use?
 - Depending on current system load, input data, architecture, ...
 - Which loop has the best granularity?
 - Depending on input data, architecture, number of threads, ...
 - Use nested parallelism?
 - Determine the optimal number of groups and their sizes

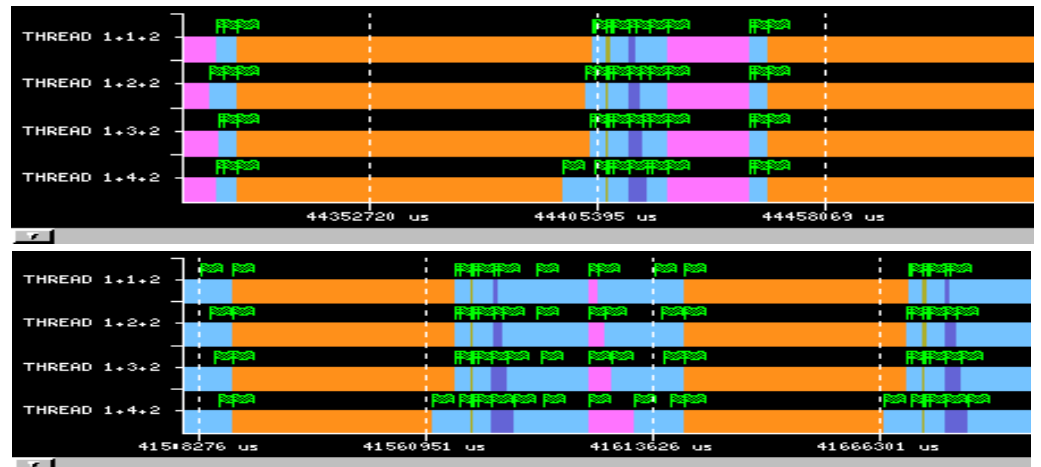
Smart scheduling

- Feedback guided scheduling
 - Load Balancing



- Architectural schedules

- SMTs
- DSMs
- ...



Conclusion

- Looking through our *crystal ball*TM
 - Parallelism goes mainstream
 - Great times for programmers :-)
 - Smart runtimes/compiler
 - Semantically powerful defaults
 - Optimized for applications & architectures
- Extra time for friends & beers