The Challenges of Irregular Parallelism

Chris Seaton
seatonc@cs.man.ac.uk

Supervisors: Ian Watson and Mikel Luján
Advanced Processor Technologies Group

Regular
Example problems
• Matrix multiplication
• Ray tracing
• Web servers

Problems where it is easy to find independent sub-tasks

Tools and techniques
• Processes, threads, actors
• Futures, sparks, fork-join
• Parallel arrays
• Dataflow

Irregular
Example problems
• Airline seat booking
• Web and social graphs
• Circuit routing

Problems where it is hard to find independent sub-tasks

Tools and techniques
• Heuristics, over-estimation
• Graph colouring
• Optimistic execution
• Transactional memory

Why don’t the regular techniques work?
They rely on knowing which tasks are independent of each other before you start the task

Why aren’t the irregular techniques enough on their own?
The regular techniques gave us important benefits that we lost with the irregular ones

• We’re doing ‘optimistic’ execution – but do we need to blindly hope for the best?
• Lost temporal and spatial locality
• Lost efficient use of cache
• We need to consider NUMA and distributed architectures

Trying a Combined Approach
Use a regular technique, and solve just the irregular parts using an irregular technique

Dataflow
Transactional memory

Tools and techniques
• Processes,
• Futures,
• Parallel arrays
• Dataflow

• Heuristics,
• Graph colouring
• Transactional memory

• We’re doing ‘optimistic’ execution – but do we need to blindly hope for the best?
• Lost temporal and spatial locality
• Lost efficient use of cache
• We need to consider NUMA and distributed architectures

Reference: