



Erlang
SF Bay Area 2012



Parallel Patterns for Multi/Many Core Architectures

Horacio González-Vélez, Ph.D.

RGU & The ParaPhrase Consortium

www.paraphrase-ict.eu

Twitter: @paraphrase_fp7



Agenda



1. Background

- The Multicore Challenge
- The Parallel Programmer's Dilemma
- Parallel Patterns and Algorithmic Skeletons

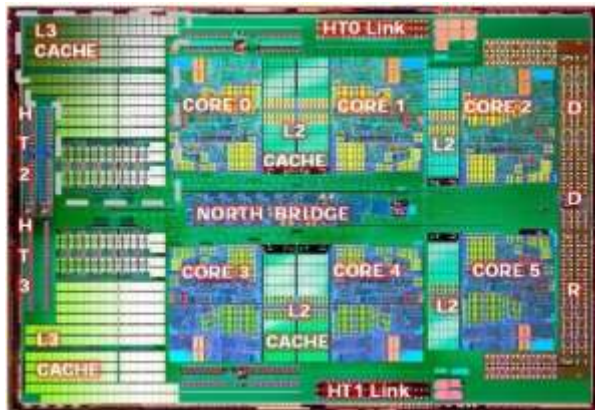
2. ParaPhrase

- Description
- Parallelisation and Virtualisation



BACKGROUND

The Dawn of a New Age?



Up to 24 cores available in AMD's Magny Cours Opteron

The Challenge



21:06

“Ultimately, developers should start thinking about **tens, hundreds, and thousands** of cores **now** in their algorithmic development and deployment pipeline.”

Anwar Ghuloum, Principal Engineer, Intel Microprocessor Technology Lab

“The dilemma is that a **large percentage** of mission-critical enterprise applications will **not** “automagically” run **faster** on multi-core servers. In fact, many will actually run **slower**. We must make it as easy as possible for applications programmers to exploit the latest developments in multi-core/many-core architectures, while still making it easy to target future (and perhaps unanticipated) hardware developments.”

Patrick Leonard, Vice President for Product Development

Rogue Wave Software

www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

5

Programming Issues



21:06

- We can muddle through on 2-8 cores
 - maybe even 16
 - modified sequential code may work
 - multiple programs to soak up cores
 - BUT larger systems are **much** more challenging
- “Think parallel”
 - New **high-level** programming constructs
 - Decouple Computation from Coordination



www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

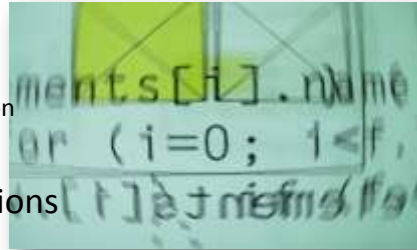
PARAPHRASE

6

'Typical' Approaches



- Applications Programmers = *Systems Programmers*
 - Insufficient assistance with abstraction
- Difficult/impossible to scale, unless the problem is simple
- Difficult to change fundamentals
 - Scheduling, Task structure, Migration
- Libraries, but we need abstractions



www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

7

Future: Flexi-Core computers?



- *Highly heterogeneous, Dark Silicon*
 - CPUs, GPUs, APUs, Soft Cores (FPGAs)
 - 100x lightweight scalar/fp units
 - 10x specialised units (graphics, auth'n, network, power control, etc)
- NUMA or even message-passing on a chip



www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

8

Implications for Programming



- Program heterogeneous systems in an *integrated* way
- *Impossible* to program each kind of core differently
- *Impossible* to make pure static decisions



Possible Language Approaches



- *Pattern-based approaches*
 - *Language agnostic*
- Parallel stream-based approaches
- Coordination approaches
- Direct programming e.g. Parallel {Haskell, Fortran, C, yours_ here}

Avoid issues such as deadlock etc...

Parallelism by Construction!

Patterns & Skeletons



- **Parallel Pattern**
 - Consists of Name, Problem Solved, Implementation Strategy
 - Use to Design.
 - Example: Embarrassingly parallel computation (ep), Staged computation (sc)

- **Algorithmic skeleton**
 - Programming Construct to implement a particular parallel pattern.
 - Use to Code.
 - Example: Farm (*ep*), Pipeline (*sc*).

www.paraphrase-ict.eu

Twitter: @paraphrase_fp7

PARAPHRASE

11

Algorithmic Skeletons



- Higher-Order Functions
- Abstract and Implement **Patterns** of Parallel Computation, Communication, and Interaction
- Decouple Behaviour (Computation) from Structure (Coordination)



Cole, M. Algorithmic Skeletons: Structured Management of Parallel Computation. Pitman/MIT Press, London, 1989.

www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

12

Algorithmic Skeletons



21:06

Skeleton	Scope	Example
Data-Parallel	Data Structures	Scan, Map, Broadcast, Reduce, Gather, Scatter,
Task-Parallel	Tasks	Farm, Pipeline, ...
Resolution	Family of Problems	Div & Conq, Br & Bnd, Dyn Prog, Heuristic Opt, ...

Gonzalez-Velez H, Leyton M. A Survey of Algorithmic Skeleton Frameworks: High-Level Structured Parallel Programming Enablers. *Software: Practice and Experience*. 2010 Dec;40(12):1135-1160. [<http>].

www.paraphrase-ict.eu

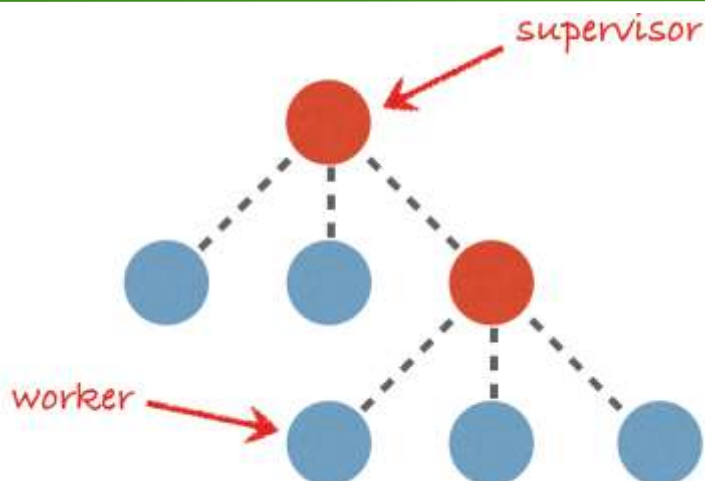
Twitter: @paraphrase_fp7

PARAPHRASE

Farms are Erlang 'Natives'



21:06



www.paraphrase-ict.eu

Twitter: @paraphrase_fp7

PARAPHRASE

14

Extend and Improve



- Standard functional *algorithmic skeletons*

```
parMap:: (a->b) -> [a] -> [b]
parZipWith:: (a->b->c) -> [a] -> [b] -> [c]
parReduce:: (a->b->b) -> b -> [a] -> b
parMapReduce:: (a->b->b) -> (c->[(d,a)]) -> c -> [(d,b)]
masterWorker:: (a->([a],b)) -> [a] -> [b]
```

- Resolution patterns (domain-specific)

- orbit calculation: generate unprocessed neighbouring states
- duplicate elimination: merge two lists
- completion algorithm: generate new objects from any pair
- chain reduction: generate new objects from any pair
- partition backtracking: search for *basis objects*
- others?? search skeleton, classification skeleton, modular skeleton+CRA, backtracking search, stencil,...

www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

15



PARAPHRASE



www.paraphrase-ict.eu

Twitter: @paraphrase_fp7

PARAPHRASE

16

Project Details



- **3 Year targeted research project (FP7 STReP)**
 - Runs from 1/10/11 to 30/9/14
 - Funded by the European Commission
- **9 partners from five countries**
 - UK, Italy, Austria, Germany, and Israel
 - Including Erlang Solutions
- **€ 3.5M**



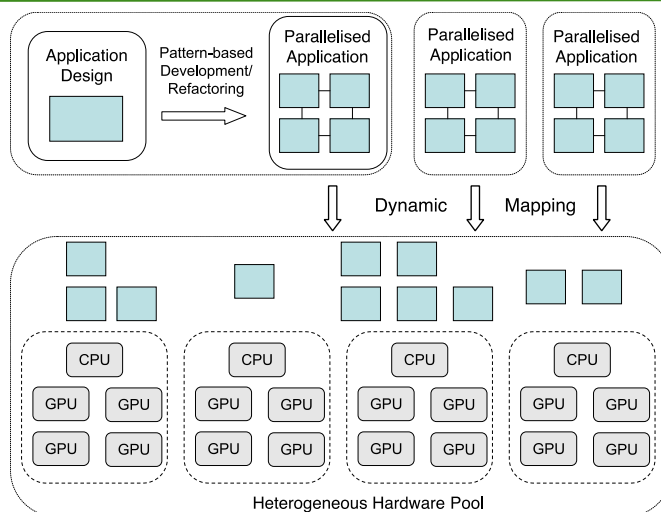
www.paraphrase-ict.eu

Twitter: @paraphrase_fp7

PARAPHRASE

17

Project Vision



www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

18

How?



- Patterns help to “think in parallel”
 - Capture structure, match implementation
- Cost-directed refactoring
 - Rewrite source to choose “best” pattern
- Virtualised *components*
 - Hardware *and* Software



- Standardised C/C++ and Erlang Patterns

www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

19

ParaPhrase



- Develop & deploy new high-level parallel patterns to be mapped/re-mapped to the available hardware.



www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

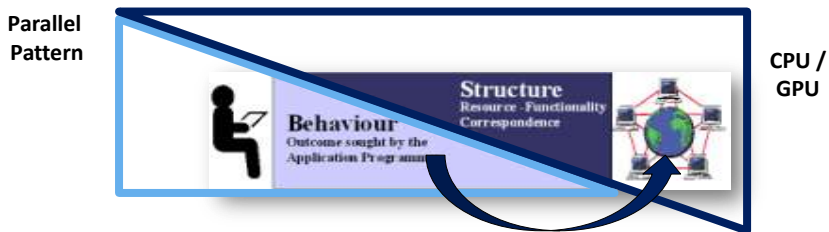
20

ParaPhrase



21:06

- Develop & deploy new high-level parallel patterns to be mapped/re-mapped to the available (**heterogeneous**) hardware.



www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

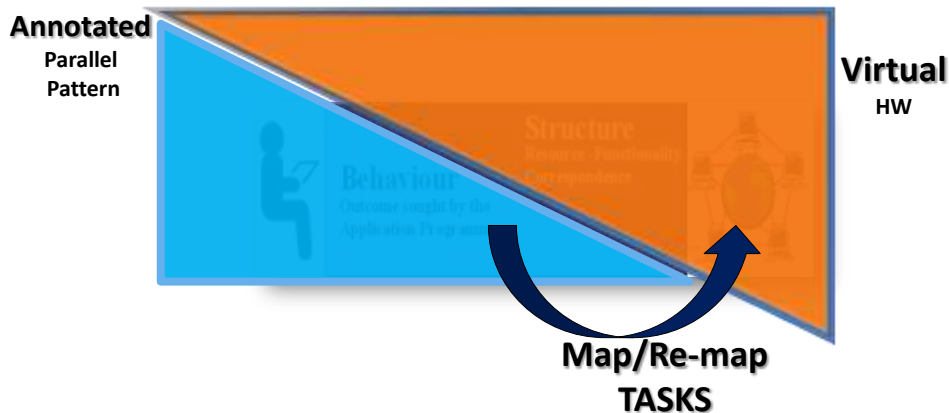
PARAPHRASE

21

ParaPhrase



21:06



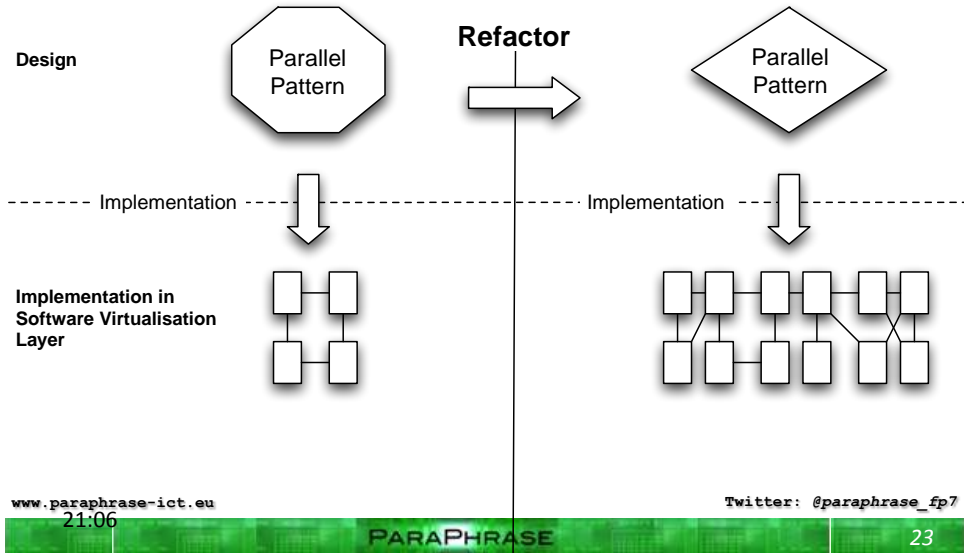
www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

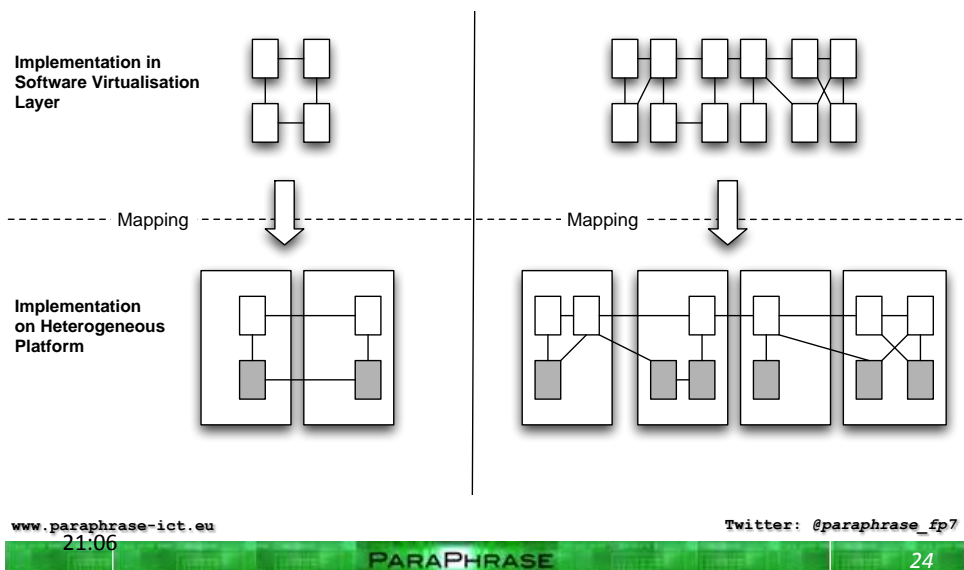
PARAPHRASE

22

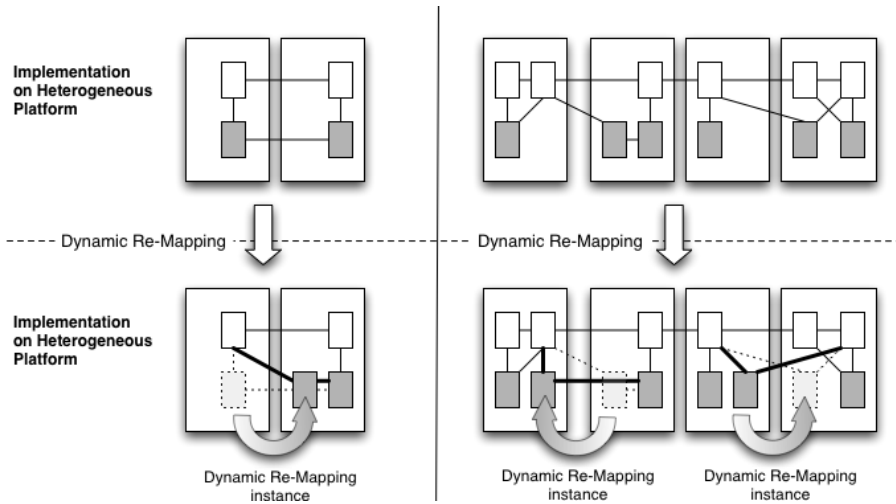
Pattern-Based Implementation



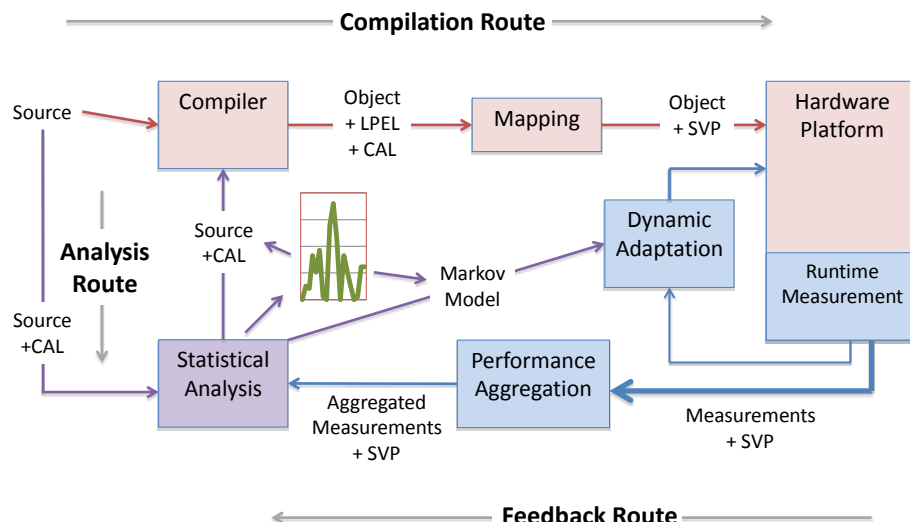
Static Mapping



Dynamic Re-Mapping



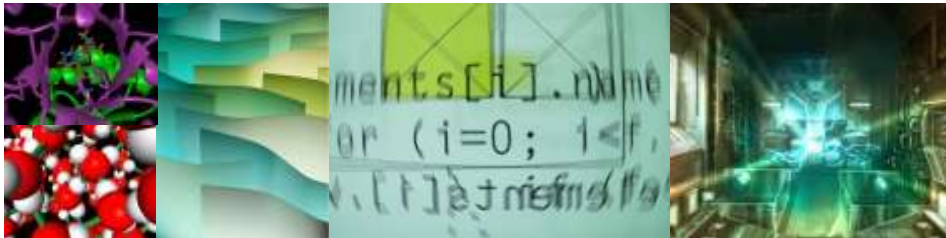
Feedback-Directed Compilation



Ultimate Objective



Integration between **computational problems** and their **parallel patterns** of computation and communication to *adaptively improve* overall **resource utilisation**.



www.paraphrase-ict.eu

Twitter: @paraphrase_fp7

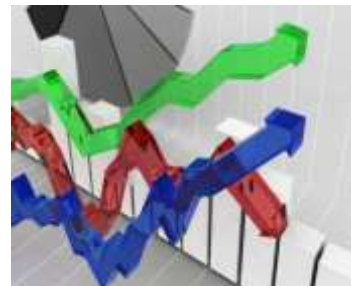
PARAPHRASE

27

Expected Long-Term Impact



- Accelerated development
 - *Dynamic* mapping code to resources via extra-functional properties.
 - *Virtualisation & Abstraction of coordination*
- Improved Speed to Market
 - *Think and Design in Parallel*
- Good Parallelism at Low-Effort



www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

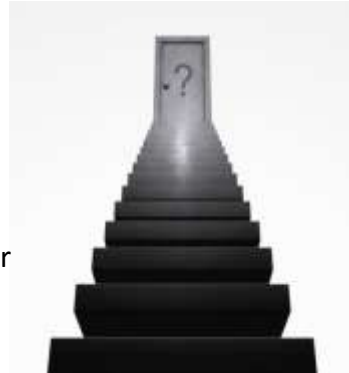
28

Research Directions



21:06

- What patterns?
 - standard , domain-specific, special for heterogeneity,...
- 'Target-less' programming?
 - what virtualisation mechanisms
 - abstract mem. access, comms, state
- What monitoring information?
 - metrics: execution time, memory, power
 - historical v. predicted information
- Static /dynamic mapping



www.paraphrase-ict.eu
21:06

Twitter: @paraphrase_fp7

PARAPHRASE

29



21:06

THANK YOU!

www.paraphrase-ict.eu
Twitter: @paraphrase_fp7

www.paraphrase-ict.eu

Twitter: @paraphrase_fp7

PARAPHRASE

30



Erlang
SF Bay Area 2012



Parallel Patterns for Multi/Many Core Architectures

Horacio González-Vélez, Ph.D.

RGU & The ParaPhrase Consortium

www.paraphrase-ict.eu

Twitter: *@paraphrase_fp7*

