

A Portable and Efficient Thread Library for OpenMP



ROYAL INSTITUTE
OF TECHNOLOGY

Sven Karlsson
Department of Microelectronics and
Information Technology, KTH
Email: Sven.Karlsson@imit.kth.se



Motivation

- The OdinMP OpenMP compiler targets a run-time library
 - Several libraries are available:
 - The Intone library
 - The Balder library
- Efficient thread primitives are vital for performance
 - The Intone library uses a non-portable thread library
 - Other compilers use POSIX threads
 - POSIX threads are not efficient
 - Balder aims at supporting multiple targets
- How can an OpenMP thread library be designed to be both efficient and portable?

Contribution

- A portable OpenMP thread library
- Performance evaluation



ROYAL INSTITUTE
OF TECHNOLOGY



Outline

- Background on Balder
- Design ideas in Balder Threads, the thread library
- Performance evaluation
- The future of Balder and OdinMP
- Summary

I won't go into details

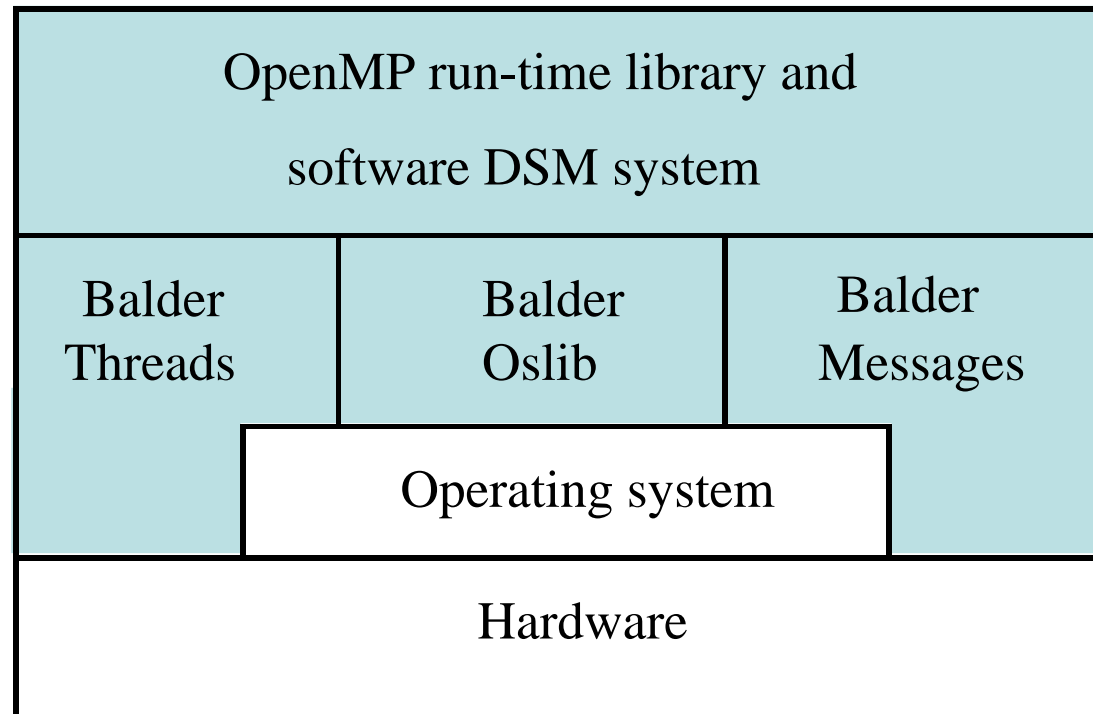
Do ask questions!



Balder

- Balder is a run-time library that OdinMP uses
- Supports OpenMP 2.0
 - Supports nested parallelism
- Also supports clusters
 - However without nested parallelism at this point
- Multiple targets
 - Linux on x86 is considered stable

Balder overview



Balder Threads

- Balder Threads provides all thread primitives used by the Balder library

Thread creation/destruction

Thread pools and work queues

Thread synchronization

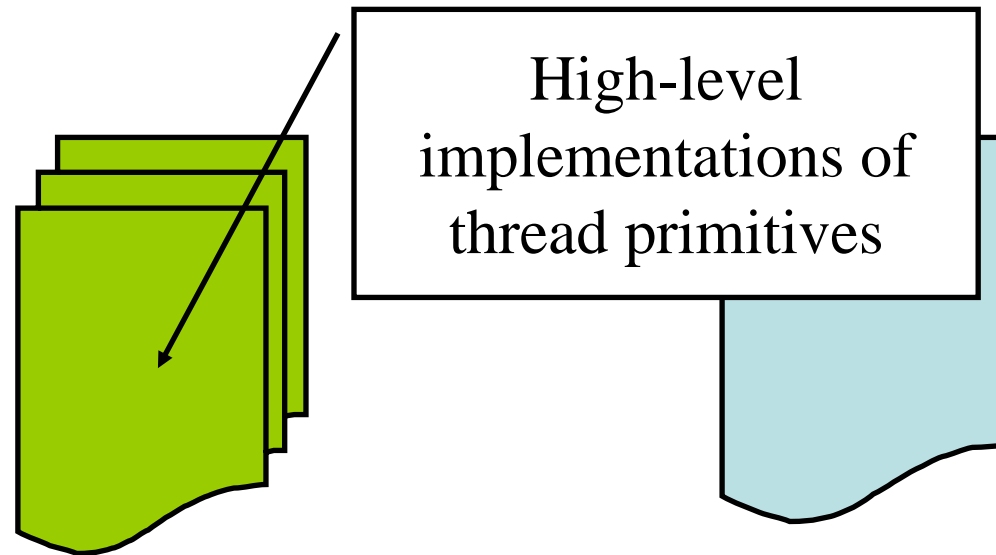
Barriers

Monitors



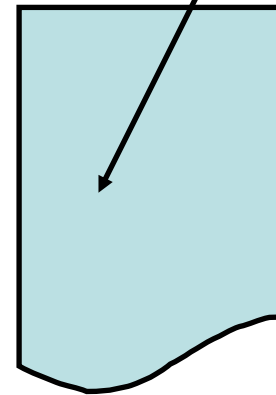
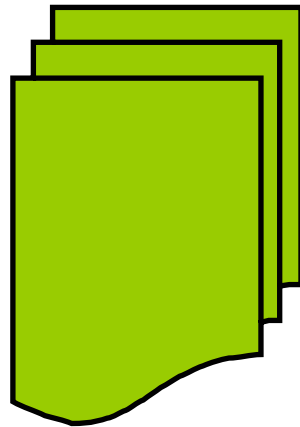
ROYAL INSTITUTE
OF TECHNOLOGY

Balder Threads, cont.



- All thread primitives in Balder Threads are written in C and are using a well defined interface to low level primitives

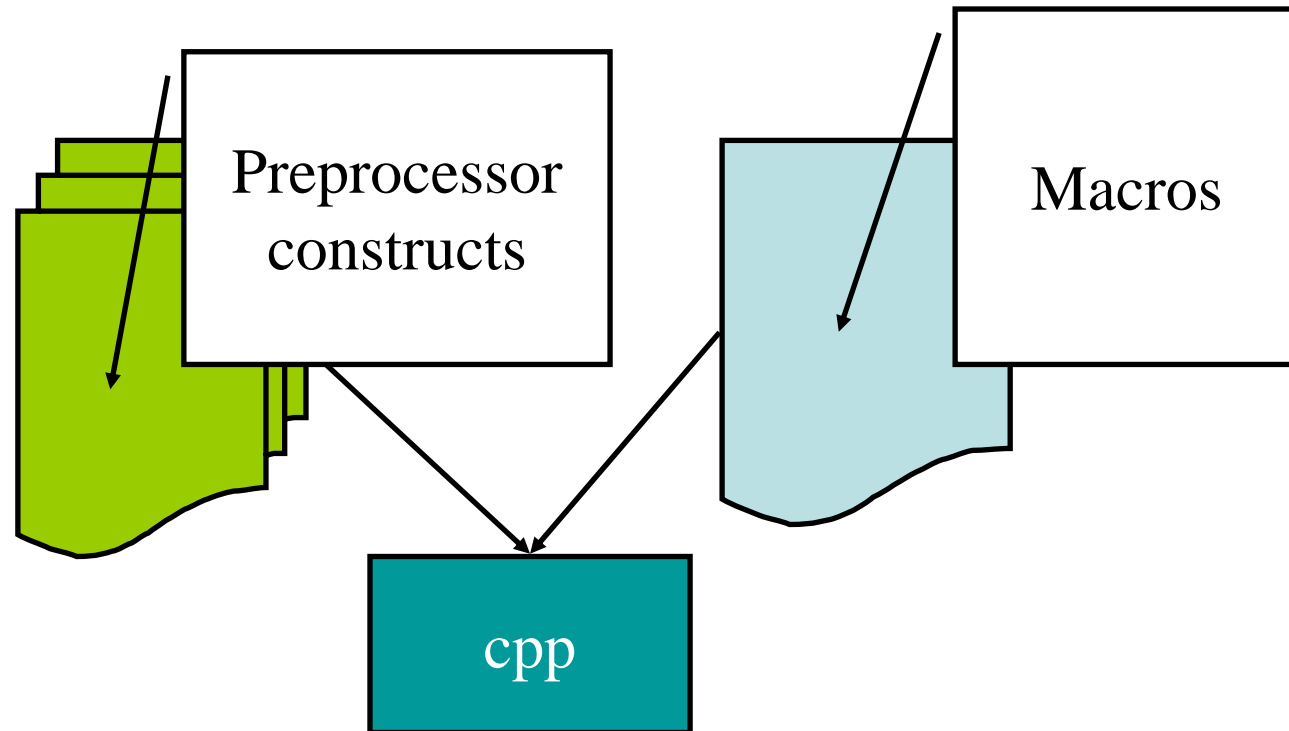
Balder Threads, cont.



Low-level
primitives
such as test-
and-set,
memory
barriers

- Each target architecture has its own set of definitions of low-level primitives
- Balder Threads falls back on default implementations if needed
Using POSIX threads

Balder Threads, cont.



- The low-level primitives are implemented as macros and combined with the high-level code during preprocessing



Experimental setup

- Dual P-III 1 GHz
Linux 2.4.25
- EPCC micro-benchmarks for OpenMP 1.0
- OdinMP 0.284.1
Balder 0.105.1
GCC 3.3.4
- Intel C/C++ compiler v. 8.0

Experimental results



ROYAL INSTITUTE
OF TECHNOLOGY

OpenMP Construct	Intel compiler	Balder with OdinMP without architecture support	Balder with OdinMP and architecture support
Parallel construct	1.43 +/- 0.11	27.15 +/- 1.35	2.91 +/- 0.15
For construct	0.79 +/- 0.17	10.59 +/- 0.87	2.93 +/- 0.30
Barrier construct	0.48 +/- 0.19	11.14 +/- 0.29	0.49 +/- 0.12
Lock and unlock primitives	0.48 +/- 0.33	6.90 +/- 0.35	0.47 +/- 0.12

- All values in microseconds and with a 95% confidence interval
- Overheads for barriers and locks more than 10 times smaller
- Overhead for parallel regions almost 10 times smaller

OdinMP and Balder news



- New domain for OdinMP and Balder
odinmp.com, find OdinMP at
<http://www.odinmp.com>
Check the website for a list of OdinMP and Balder related papers
You will find these slides there
- OdinMP supports OpenMP 2.0
Nested parallelism and ANSI C99 fully
2.0 version not yet released
Work on 2.5 will start when I receive a draft of the specification!
- Balder will soon be released
The 0.106.1 release and distribution are done,
ready, and available on request
Awaiting license issues
Will not include cluster support
Linux on x86



ROYAL INSTITUTE
OF TECHNOLOGY

Future plans

- OdinMP will support Fortran!
Fortran 77/90/95 and 2003 (when it arrives)
Effort just began
- The C/C++ parser will be overhauled
Aiming at full ANSI C++ support and better support for legacy C
- OdinMP will receive support for some C/C++ extensions
Already support for some GCC extensions
Support for Microsoft's compilers planned



Future plans, cont.

- Multiple targets for Balder are in the works
 - Linux on ia64
 - Linux on Sparc
 - Solaris on Sparc
 - Win32 on x86
- Public release of cluster support is being prepared
- Let me know if you want anything in particular
 - I don't have access to each and every architecture there is
 - Accounts, donation of CPU time and other resources is very much appreciated

Summary



ROYAL INSTITUTE
OF TECHNOLOGY

- A portable and efficient thread library is discussed
- Overheads are an order of a magnitude smaller than POSIX threads and comparable to commercial implementations
- OdinMP is now 2.0 compliant
- Fortran version of OdinMP is planned
- Balder is to be released

Thank you!

- Questions?



**ROYAL INSTITUTE
OF TECHNOLOGY**

Overview of the compilation system

