# Towards an Efficient Fault-Tolerance Scheme for GLB

Claudia Fohry, Marco Bungart and Jonas Posner

#### U N I K A S S E L V E R S I T A T



Programming Languages / Methodologies

イロト イポト イヨト イヨト

June 14, 2015

UNIKASSEL VERSITÄT Fault Tolerance Scheme

Experimental Results

# Global Load Balancing



- Pault Tolerance Scheme
- 3 Experimental Results

UNIKASSEL VERSITÄT



Fault Tolerance Scheme

Experimental Results

3/18

イロト イポト イヨト イヨト

#### Worker-local Pools



Examples:

- UTS: counting nodes in an unbalanced tree
- BC: calculate a property of each node in a graph

U N I K A S S E L V E R S I T 'A' T

#### GLB

- Task pool framework for inter-place load balancing
- Utilizes cooperative work stealing
- Tasks are free of side effects and can spawn new task at execution time
- Final result computed by reduction
- Only one worker per place
- Worker-private pool





5/18

イロト イポト イヨト イヨト

# GLB's main processing loop

```
do {
    while (process(n)) {
        Runtime.probe();
        distribute();
        reject();
     }
} while (steal());
```

U N I K A S S E L V E R S I T A' T Fault Tolerance Scheme

Experimental Results

#### Fault Tolerance Scheme







U N I K A S S E L V E R S I T 'A' T



イロト イポト イヨト イヨト

#### Conceptual Ideas

- One backup-place per place (cyclic)
- Write backup periodically and when necessary (stealing)
- Exploit stealing-induces redundancy
- Write incremental backups whenever possible
- Each information at exactly two places



Experimental Results

8/18

#### Incremental Backup of stable Tasks



9/18

イロト イポト イヨト イヨト

# Actor Scheme

- No blocking constructs (except one outer finish)
- split and merge have to operate on the bottom of the Task Pool
- Actor Scheme
  - Worker is passive entity (only processing tasks)
  - Worker becomes active when a message is received
  - Two kinds of messages:
    - executed directly or
    - stored and processed later
- $\rightarrow$  Worker stays responsive

```
U N I K A S S E L
V E R S I T 'A' T
```

# Stealing Protocol





U

イロト イヨト イヨト イヨト э 10/18

U

V

Fault Tolerance Scheme

Experimental Results

#### Asynchronism



11 / 18

#### Asynchronism with Fault-Tolerance



#### Detection of dead Places

- Cannot use DeadPlaceExceptions
- Check relevant places regularly via isDead(), as well as the own backup place
- What if a place P is inactive?
  - Does not check its backup-place for lifeness
  - But its predecessor Forth(P) does check P
  - If P is active, it checks lifeness of Back(P)
  - Recursive process





Experimental Results

#### **Experimental Results**







U N I K A S S E L V E R S I T 'A' T



#### Setup

Experiments were conductet on an Infiniband-connected Cluster

- One place per node
- Up to 128 Nodes
- Configuration:
  - small UTS: -d=13
  - large UTS: -d=17





**GLB** 

Fault Tolerance Scheme

Experimental Results

# UTS, small



うへで 16/18

U

v

Fault Tolerance Scheme

# UTS, small



୍ରର୍ ୧୦ 17/18

# Thank you for your attention!

Please feel free to ask questions.



