Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating Examples

Example 1 Example 2

$\mathsf{Problem}$

CCD

New Scheduling

Rel-Acq Dependencies

New Schedulin

Evaluation

Pros/Cons

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Nadesh Ramanathan, John Wickerson, George Constantinides.

Presented by Akshay Gopalakrishnan

February 17, 2022

Who are they?

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Yann Herklotz -PhD student supervised by John Wickerson. Imperial College London.

John Wickerson -Lecturer at Dept of Electrical and Electronic Engineering, Imperial College London

George Constantinides -Professor of Digital Computation, Imperial College London.







Scheduling Code block 1

int r0=0, r1=0, r2=0; r0=y+y+y+y+y;

r1=x;

r2=x/a:

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivatir Examples

Example 1

B 11

SC Dependency

New Schedulin

Dependencies

New Scheduli

Evaluation

1 2 3	4 5 6	7	36
-------	-----------	---	----

ld y										
	ld y									
	ld y									
	ld y	7								
	ld y	7								
		ld								
		ld	Х							
ld x										
	divide									

Scheduling Code block 2

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrish-

Motivatin Examples

Example 1

Duckley

Donondo

SC Depende

Rel-Acq

Dependencies

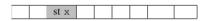
Evaluation

Evaluation

Pros/Cons

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... | 36 |

x=1;



Concurrent scheduling of Block 1 and 2

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating

Example 1

Probler

SC Dependen

Rel-Acq Dependencies

Evaluation

MP Example Pros/Cons int r0=0,r1=0,r2=0;
r0=y+y+y+y+y;
r1=x;
r2=x/a;

x=1:

 $assert(r1 = 1 \Rightarrow r2 \neq 0)$

1 2 3 4 5 6 7 36	5 6 7 · · ·	36
----------------------------------	-------------------	----

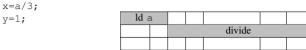
ld y					
ld y					
ld y					
ld	У				
ld	У				
	1d	У			
	ld	х			
ld x					
			livid	c	



Scheduling Code block 1

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

1	2	3	4	5		35	36
---	---	---	---	---	--	----	----



st y



st x

Scheduling Code block 2

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivatii Examples

Example 1

Probler

SC Depende

New Scheduli

Dependencies

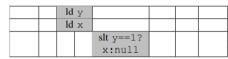
Evaluation

MP Example

MP Exampl Pros/Cons

1 2 3 4 5 35 36	35 36
-------------------------------	---------

Inc 10-0,1	1-0;
r0=y;	
if(r0==1)	r1=x;



Concurrent scheduling of Block 1 and 2

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivatin Examples

Example 1

Example 2

- TODIEI

SC Dependent

Rel-Acq Dependencies

Evaluation

MP Example Pros/Cons

ı	1	2	3	4	5	 35	36
н	•	_	_				-

$$assert(r0 = 1 \Rightarrow r1 = 1)$$

ld a	a								
		divide							
						st x			
st y									
st y									

	ld	У			
	ld	х			
			slt $y==1$?		
			x:null		



Data Dependencies: non-aliasing

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating Examples

Example 1

Proble

Dependency SC Depende

New Scheduli

Rel-Acq Dependent

Dependencies New Schedulii

Evaluation

$$E_{\text{intra-iter}} = \{(v, v', 0) \mid sb(v, v') \land sloc(v, v') \land (v \in V_{\text{st}} \lor v' \in V_{\text{st}})\}$$

$$E_{\text{inter-iter}} = \{(v, v', 1) \mid sloc(v, v') \land (v \in V_{\text{st}} \lor v' \in V_{\text{st}})\}.$$

Adding WW—WR—RW—RR Dependancies

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating Examples

Example 1 Example 2

Dependen

SC Dependencies New Scheduling

Rel-Acq Dependencies New Schedulii

Evaluation

MP Example Pros/Cons

```
\begin{split} E_{\mathrm{sc} \ddagger} &= \{(v, v', 0) \mid sb(v, v') \land v \in V_{\mathrm{sc}}\} \\ E_{\mathrm{sc} \ddagger} &= \{(v, v', 0) \mid sb(v, v') \land v' \in V_{\mathrm{sc}}\} \end{split}
```

Final Dependency Expression

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

SC Dependencies



Example

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating Examples

Example 1

Probler

SC Dependencie

Rel-Acq Dependencies

Evaluation

MP Example Pros/Cons

Without pipelining

Cycle:	1	2	3	4	5	6	7	8	9	10	11	12
r1=x;	ld	ld _{na} x					ld _{na} x					
r2=ld(&y,ACQ);		ld _{ACQ} y						ld	ACQ Y			
r0=z;					ld	na Z					ldna	a Z

With pipelining

Cycle:	1 2 3 4		5	6	7 8		9	10		
r1=x;	ld _{na} x				ld _{na} x					
r2=ld(&y,ACQ);	ld _{ACQ} y					ldA	со у			
r0=z;				ld _{na} z				ldn	a Z	

Weakening: Adding Release-Acquire Dependencies

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Rel-Aca Dependencies

$$E_{\text{acq} \downarrow} = \{(v, v', 0) \mid sb(v, v') \land v \in V_{\text{acq}}\}$$

$$E_{\text{rel} \uparrow} = \{(v, v', 0) \mid sb(v, v') \land v' \in V_{\text{rel}}\}$$

Adding RR Dependency

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Rel-Aca Dependencies

```
E_{\text{RAR}} = \{(v, v', 0) \mid sb(v, v') \land sloc(v, v') \land slo
                                                                                                                                                                                                                                                                                                                                                                                                 v \in V_{\rm at} \cap V_{\rm ld} \wedge v' \in V_{\rm at} \cap V_{\rm ld} \}.
```

Final Dependency Expression

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating

Example 1

Example 2

Problem

SC Dependent

Rel-Acq Dependencies

Evaluation

Pros/Cons

```
\begin{split} E_{\text{mem,weak}}^{\text{pipe}} &= E_{\text{intra-iter}} \cup E_{\text{inter-iter}} \cup E_{\text{sc} \updownarrow} \cup E_{\text{sc} \updownarrow} \cup \\ & E_{\text{acq} \updownarrow} \cup E_{\text{rel} \updownarrow} \cup E_{\text{RAR}} \cup \\ & E_{\text{sc-inter-iter}} \cup E_{\text{acq-inter-iter}} \cup \\ & E_{\text{rel-inter-iter}} \cup E_{\text{RAR-inter-iter}} \end{split}
```

Example

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating Examples

Example 1 Example 2

Probler

SC Dependenci

Rel-Acq Depend

New Scheduling

Evaluation

MP Example

MP Example Pros/Cons

Without pipelining

Cycle:	1	2	3	4	5	6	7	8
r1=x;	ld _{na} x				ld _{na} x			
r2=ld(&y,ACQ);	ld _{ACQ} y				ld _{ACQ} y			
r3-z;			ld _{na} z				ld,	na Z

With pipelining

Cycle:	1	2	3	4	5	6
r1=x;	ld _{na} x		ld _{na} x			
r2=ld(&y,ACQ);	ld _{ACQ} y		ld _A	со У		
r3=z;			ld _{na} z		ld _{na} z	

Message Passing Algorithm

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating Examples

Example 1

Problen

SC Dependenci

Rel-Acq Dependencies

Evaluation

```
atomic_int flag1 = 0, ..., flagN = 0;
              int data<sub>1</sub> = 0, ..., data<sub>N</sub> = 0;
in for (i=0; i<ITER; i++) { | 21 for (i=0; i<ITER; i++) {
    if(ld(&flag_1,ACO)==0){
                                22 if (ld(&flag1,ACO) ==1){
     data1++;
                                      data<sub>1</sub>++;
     st(&flag1,1,REL);
                                    st(&flag1,0,REL);
    if (ld(\&flag_N, ACQ) == 0) { ||2.7| if (ld(\&flag_N, ACQ) == 1) {
     datav++:
                                      data_N++;
1.8
     st(&flagN, 1, REL);
                                      st(&flagN, 0, REL);
1.10
1.11
```

Fig. 5. A two-threaded message-passing example with acquire-release semantics on N independent channels.

Impact of Modified Scheduling

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating

Example 1

Problen

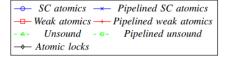
SC Dependenc

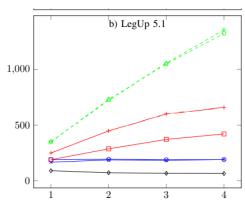
New Schedulin

Dependencies

Evaluation

Lvaidation



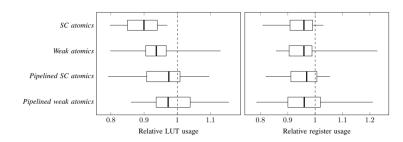


Number of independent channels, N



Resource Usage

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware



Positives

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrish nan

Motivating Examples

Example 1 Example 2

Probler

SC Dependenc

Rel-Acq Dependencies

Evaluation

MP Example
Pros/Cons

- New scheduling for programs with atomics.
- Do not require locks/notion of critical section for Hardware design.
- Efficient and correct scheduling

Pending Work

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented b Akshay Gopalakrish nan

Motivating Examples

Example 1
Example 2

Problen

SC Dependen

Rel-Acq Dependencies

Dependencies New Schedulir

Evaluation

MP Example

MP Example Pros/Cons

- Does not support Atomic Read-Modify-Write instructions.
- Addressing other potential transformations (eg: Elimination, Introduction, Inlining, etc) for efficient scheduling.

Thank you

Scheduling Weakly Consistent C Programs for Reconfigurable Hardware

Presented by Akshay Gopalakrishnan

Motivating

Example 1 Example 2

Problen

SC Dependent

Rel-Acq Dependencie

Dependencies New Scheduli

Evaluation

Pros/Cons

- John's Blog
- Previous Work
- Testing+Verification

Questions?