SCOOP v1.5.5

Manual

September 30, 2013
(Not fully up to date)

Author: *Foivos S. Zakkak*

Foundation for Research and Technology - Hellas (FORTH)
Institute of Computer Science
N. Plastira 100
Vassilika Vouton, GR-700 13 Heraklion, Crete, Greece

License

Foivos Zakkak <zakkak@ics.forth.gr>
Dimitris Chassapis <polyvios@ics.forth.gr>
Polyvios Pratikakis <polyvios@ics.forth.gr>

FORTH-ICS / CARV
(Foundation for Research & Technology -- Hellas,
Institute of Computer Science,
Computer Architecture & VLSI Systems Laboratory)

Contents

1 Annotation Syntax

```
#pragma css start(list of variables)
#pragma css finish
#pragma css malloc
#pragma css free
#pragma css sync
#pragma css wait all
#pragma css wait on(list of variables) //Not implemented yet
#pragma css task [in(<input parameters>)]
                 [inout(<input parameters>)]
                 [out(<input parameters>)]
                 [highpriority] (highpriority is simply ignored)
```

Parameter notation:

Non stride: `<parameter>[\[number of elements (for arrays)\]]` [1]

Stride: `<parameter>\[Block Rows|Block Columns\]\[[Array Rows|]Array Columns\]`
Array Rows is optional and is totally ignored

Stride (Legacy): `<parameter>\[stride\]\[ els | elsz \]`
where stride is the step (original row length)
els is the block's number of rows
elsz is the block's row length

The parameter size/stride/els/elsz must be an expression, thus we don't allow function calls. Also there is no support for the conditional operator (? :)

Example: `#pragma css task in(a, b[4]) out(c[16])`

2 Installing

This section describes how to install SCOOP on your system. We suppose that you have checked out/cloned SCOOP under `/opt/scoop` directory. However the same instructions should apply for any alternative path, simply replacing `/opt/scoop` with the desired alternative path for the rest of this section.

2.1 Dependencies

In order to build SCOOP you will need to install the following packages:

[1] XPPFX and nesting require the size in bytes and not in elements

- ocaml
- camlp4/ocaml-camlp4/ocaml-camlp4-devel
- flex
- bison
- indent
- ncurses-devel
- emacs
- gperf

for *CELL B.E* runtimes you will also need the cell development package including ppu_intrinsics.h, altivec.h etc in your include path (C_INCLUDE_PATH)
(e.g. `export C_INCLUDE_PATH=$C_INCLUDE_PATH:/opt/cell/toolchain/lib/gcc/ppu/4.1.1/inclu`
`export C_INCLUDE_PATH=$C_INCLUDE_PATH:/opt/cell/toolchain/lib/gcc/spu/4.1.1/inclu`
and ppu32-gcc in your PATH
(e.g. `export PATH=$PATH:/opt/cell/toolchain/bin`)

2.2 Compile

To compile SCOOP you have to run `configure` and then `make`.

Code

```
$   ./configure && make
```

2.3 Install

You can install SCOOP running

Code

```
$   sudo make install
```

this will create a copy of the scoop executable in `/usr/local/bin`.

Alternatively you can append `/opt/scoop` to the `PATH` variable.
i.e.

Code

```
$   echo "export PATH=$PATH:/opt/scoop" >> $HOME/.bashrc
```

2.4 Uninstall

You can uninstall SCOOP running

Code

```
$   sudo make uninstall
```

this will erase the copy of the scoop executable from `/usr/local/bin`.

If you chose the alternative method of adding `/opt/scoop` to your `PATH` variable, simply remove the added line from your `.bashrc`

3 Using

Code

```
$   scoop --runtime=<adam/bddt/cell/cellgod/cellBlade/cellgodBlade/myrmics/scc/nesting/XPPFX> [optio
```

3.1 Options (Needs update)

`--runtime`	Defines the target runtime/architecture adam — bddt — cell — cellgod — cellBlade — cellgodBlade — myr
`--cflags`	Defines the flags you want to pass to gcc
`--tpcIncludePath`	Defines the include path for the tpc runtime
`--debugSCOOP`	Print debugging information
`--trace`	Trace SCOOP
`--pragma`	Specify the string constant following the pragma e.g. (default: css) w
`--out-name`	Specify the output files' prefix. e.g. (default: scoop_trans) will produ (and scoop_trans_func.c for cell)
`--queue-size`	Specify the queue size for Cell. Defined in the Makefile as MAX_QUEUE_ENTRIES
`--with-stats`	Enable code generation for statistics, for use with -DSTATISTICS
`--threaded`	Generate thread safe code for Cell, for use with -DTPC_MULTITHREADED
`--disable-sdam`	Disable the static dependence analysis module

4 Common Errors/Limitations/Known Bugs

- Adding a semicolon at the end of `#pragma`s
- Putting `#pragma css barrier` at the end of a block (fix, add a semicolon right below of it)
- **Fatal error: exception Invalid_argument("Unknown")** you probably have wrong argument at a call tagged with `#pragma css task`
- Using `#define`s in pragmas (preprocessor doesn't process them)
- Putting `#pragma`s directly above a declaration of a variable (pragmas are only supported above statements)

- **Error: "segment___0" not found in the #pragma css task** usually means that the tool is renaming a variable due to previous declaration try renaming it manually (e.g. segment2) (This should be fixed by now)
- Using `tpc_wait_all` instead of `#pragma css barrier` results in broken SDAM