UNIVERSITÀ DI PISA

Scuola Superiore
Sant'Anna
di Studi Universitari e di Perfezionamento

A Framework for static allocation of parallel OpenMP code on multi-core platforms

Giacomo Dabisias, Filippo Brizzi

Università degli studi di Pisa,
Scuola Superiore Sant'Anna
Pisa,Italy

February 28, 2014

Context and motivations

Real-time systems are moving towards multicore architectures. The majority of multithread/core libraries target high performance systems.

- Real-time applications need strict timing guarantees and predictability.

Vs

- High performance systems try to achive a lower computation time in a best efford manner.

There is no actual automatic tool which has the advantages of HPC with timing contrains.

Objectives and Design Choice: OpenMP and Clang

OpenMP

- Minimal code overhead.
- Well spread standard.
- Opensource and supported by several vendors like Intel and IBM.

Clang

- Provides code analysis and source to source translation capabilities.
- Modularity and great efficency.
- Opensource and supported by several vendors like Google and Apple.

OpenMP

Multiple threads of execution perform tasks defined by directives.

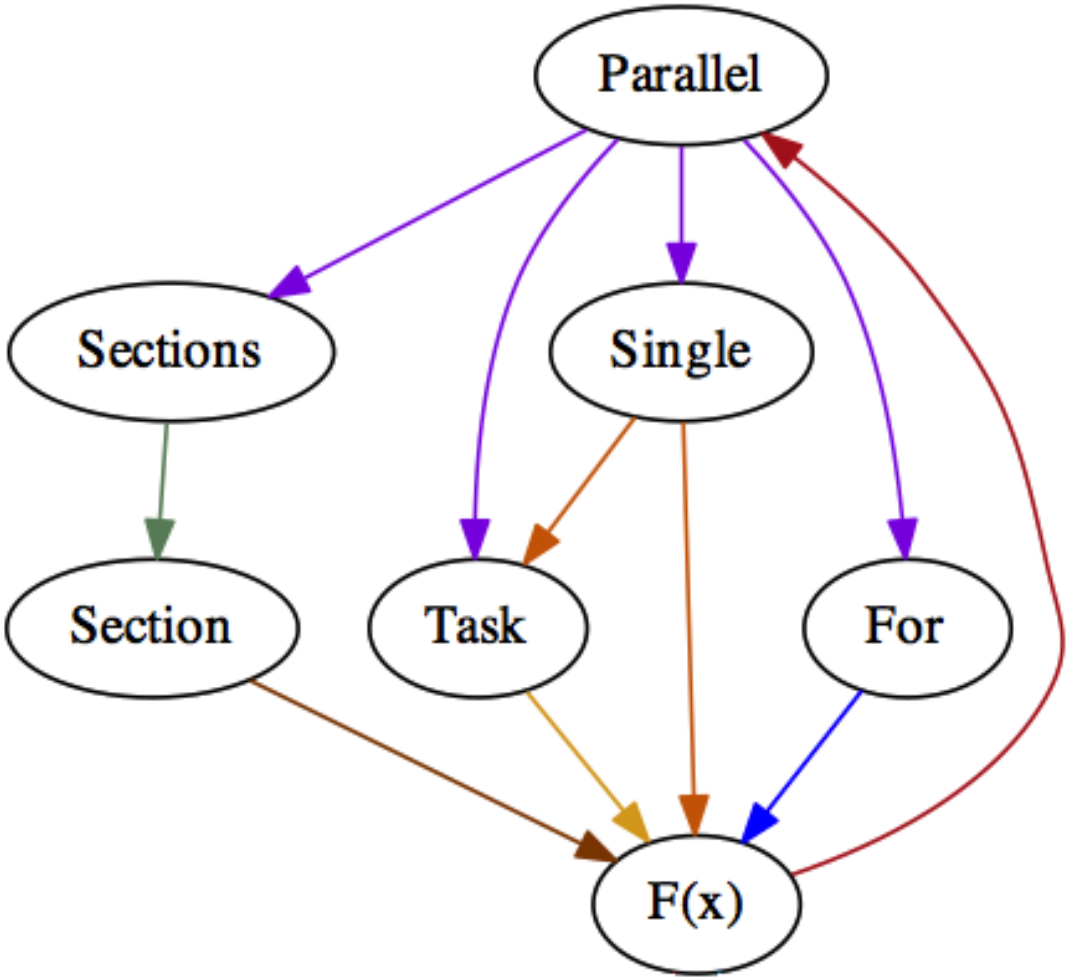
- Each directive applies to a block of C++ code embedded in a scope.
- Allows nested parallelism though nested directives.
- Clauses allow variables management.

```
#pragma omp directive-name [clause[ [,] clause]...] new-line
```

Choosen subset for the framwork:

- Control directives : parallel, sections, single.
- Working directives : task, section, for.

OpenMP



Clang

Clang and OpenMP:

- In July 2013 Intel released a patched version of Clang which fully supports the OpenMP 3.3 standard.

The strength of Clang lies in its implementation of the Abstract Syntax Tree (AST).

- Closely resembles both the written C++ code and the C++ standard.
- Clang's AST nodes are modeled on a class hierarchy that does not have a common ancestor.
- Hundreds of classes for a total of more than one hundred thousand lines of code.

Clang - AST

To traverse the AST, Clang provides the RecursiveASTVisitor class.

- Very powerful and easy to learn interface
- Possibility to create a custom visitor that triggers only on specific nodes.

Clang supports the insertion of custom code through the Rewriter class.

- Allows insertion, deletion and replacement of code.
- Operations are performed during the AST visit.
- A new source file with all the modifications is generated at the end of the visit.

Clang - AST

```
1  class A {
2  public:
3    int x;
4    void set_x(int val)
       {
5          x = val * 2;
6    }
7    int get_x() {
8        return x;
9    }
10 };
11 int main() {
12     A a;
13     int  val = 5;
14     a.set_x(val);
15 }
```

```
TranslationUnitDecl
|-CXXRecordDecl <clang_ast_test.cpp:2:1, line:13:1>
      class A
| |-CXXRecordDecl <line:2:1, col:7> class A
| |-AccessSpecDecl <line:3:1, col:7> public
| |-FieldDecl <line:4:2, col:6> x 'int'
| |-CXXMethodDecl <line:5:2, line:7:2> set_x 'void (
    int)'
| | |-ParmVarDecl <line:5:13, col:17> val 'int'
| | `-CompoundStmt <col:22, line:7:2>
| |   `-BinaryOperator <line:6:3, col:13> 'int' lvalue
      '='
| |     |-MemberExpr <col:3> 'int' lvalue ->x
| |     | `-CXXThisExpr <col:3> 'class A *' this
| |     `-BinaryOperator <col:7, col:13> 'int' '*'
| |       |-ImplicitCastExpr <col:7> 'int' <
    LValueToRValue>
| |       | `-DeclRefExpr <col:7> 'int' lvalue ParmVar
      'val' 'int'
| |       `-IntegerLiteral <col:13> 'int' 2
...
```

The framework

General Design

Big-graph image

Simple example

Pragma extraction

Intrumentation for profiling

Intrumentation for profiling - Annotated example

Flow graph


IN SUPREMÆ DIGNITATIS
1343


SCUOLA SUPERIORE SANT'ANNA
PISA

Scheduler


IN SUPREMÆ DIGNITATIS
1343


SCUOLA SUPERIORE SANT'ANNA
PISA

Scheduler - Search tree

Scheduler - Constraints check

Scheduler - (Cetto & Chetto)

Final execution

Final execution - Intrumentation

Final execution - Run-time

Fianl execution - (thread pool)

Final execution - (multiple job queues)

Final execution - (synchronization)

General structure

General structure -(graph of the test code)

Results


IN SUPREMÆ DIGNITATIS
1343


SCUOLA SUPERIORE SANT'ANNA
PISA

Results - (some tables and graphs)

Results - (total completion time)

Results - (service time - boxplot)

Results - (Jitter)

Results - Comments


IN SUPREMÆ DIGNITATIS
1343


SCUOLA SUPERIORE SANT'ANNA
PISA