draft outline for ipp

Sheng Li

<2019-02-22 Fri 13:39>

Contents

1 motivation

1. deterministic patterns -> higher maintainability [1]
    - less race conditions -> **determinism and consistency** -> simplify debugging and testing
2. application oriented patterns -> higher productivity [1]
    - patterns derived from common use cases in applications
        - universal patterns
        - patterns that target specific domains
    - encourage high-level reasoning
        - focus users on parallelism and data locality
        - simplify learning to write efficient programs

The motivation of this project is to simplify the challenge of writing efficient and correct parallel programs.

- extend an existing library
- describe general subject area

[1] McCool, parallel book

1.1 problem statement

1.2 research hypothesis and objectives

1. design
2. implement
3. test on examples
4. refactor and optimize:

    (a) refactor/what is refactoring?

    - fulfill functionality,
    - make code more readable and maintainable,
    - hide parallelism behind the interface

    (b) optimize: means improving performance of functionality

characteristic	way to optimize
neighbour structure	data reuse and data locality
regular structure of memory reads	vectorized elemental function can use shifts
multi-dimensional stencil	cache

5. benchmark

    -

1.3 timeliness and novelty

1.4 significance

1.5 beneficiaries

2 background

- existing systems: 3 categories parallel systems classified by level of abstraction
    - advantages: simple interface
    - disadvantages: sacrifice freedom
- 2 issues:
    1. problem decomposition

    - identify parallelism:
        - describe processes in a way that they can operate concurrently to achieve one goal
        - identify computations that can be preformed at the same time

2. distribution
    - specify a mapping from concurrent operations to processors (can be dynamically)
3. code and data sharing

3 programme and methodology

4 evaluation

5 expected outcomes

6 research plan, milestones and deliverables

- ☒ gantt chart
- □ milestones
- □ deliverables

6.1 gantt chart

1. □ design: prototype skeleton
2. □ 0th implement/test: prototype skeleton
    - □ implement: in doing so, I have to implement test at the same time. So I combine them.
    - □ test: implement examples
        - □ choose 1 example, candidates: Jacobi, image filtering
        - □ implement equivalent sequential program for 1st example to test correctness
3. □ 0th refactor/optimize:
    - possible refactoring/optimizations at 0th round

  - □ interface

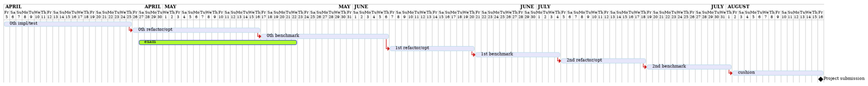
4. □ 0th benchmark:
   - □ implement equivalent naive parallel program, use pthread or use thread only

5. □ 1st refactor/optimize:
   - possible refactoring/optimizations at 1st round:
     - □ interface
     - □ shifts
     - □ granularity: workload
     - □ dynamic/static job allocation
     - □ scheduling

6. □ 1th test:
   - implement sequential program for 2nd example

7. □ 1th benchmark:
   - implement naive parallel program for 2nd example

8. □ got to stage 5



6.2 milestones

Milestone #	Week	Description
M1	W17	End of 0th implementation/test
M2	W23	End of 0th cycle
M3	W27	End of 1st cycle
M4	W31	End of 2nd cycle
M5	W33	Submission of code and dissertation

- ipp tut4: title, motivation/purpose, completion criteria, impact