Parallel Programming Course Development based on Parallel Computational Thinking

Juan Chen
College of Computer, National University of Defense Technology
Changsha, Hunan, China
juanchen@nudt.edu.cn

Jianping Yin
College of Computer, National University of Defense Technology
Changsha, Hunan, China
jpyin@nudt.edu.cn

Li Shen
College of Computer, National University of Defense Technology
Changsha, Hunan, China
lishen@nudt.edu.cn

Chunyuan Zhang
College of Computer, National University of Defense Technology
Changsha, Hunan, China
cyzhang@nudt.edu.cn

ABSTRACT
Teaching and training for high-performance computing in our college could not catch up with HPC research level. Thus, it is imperative to promote teaching reform on parallel computing course in our college. Our first parallel programming course is mainly for the first-grade graduate students majoring in CS and related branches with no previous HPC training. The goal is to teach them basic parallel programming methods, parallel thinking and parallel problem solving methodology by coding on a real supercomputer; let the students learn some representative parallel application development issues and some big challenges in HPC by project practice. In this article, we will present our course design objective, principles, practical method and outcome. Particularly, programming practice methodology, project organization, incentive mechanism and assessment methods in project practice will be illustrated. Finally, we present some quantitative findings. According to the feedback, our first parallel programming course achieves the effectiveness on inspiring students’ enthusiasm for programming and improving students’ abilities for problem solving.

CCS CONCEPTS
• Computers and Education → Computer and Information Science Education; • Programming Techniques → Concurrent Programming;

KEYWORDS
Parallel Programming, HPC and Project Practice

1 INTRODUCTION
Chinese supercomputer system design and research level have made great progress in these years. The supercomputer built by our college has had a very strong competition ability in the recent TOP500 supercomputer ranking. However, in our college, the course design, teaching method for high-performance computing (HPC) related courses are behind the international advanced level. There is a big gap between our HPC research and HPC education. Therefore, currently our courses could not meet the demands for HPC talents. It is imperative to promote teaching reform on the first parallel programming course, which is mainly for the first-grade graduate students who are majoring in CS and related branches with no previous parallel computing training.

Nowadays, many efforts on developing parallel computing related courses have been done in the world. Many educators think parallelism is the heart of the CS curriculum and they care about how to integrate parallel computation to CS curriculum [2] [3] [4]. Some research is about parallel programming course practice [5] [8]. For example, [8] presents a 16 week course in parallel and concurrent programming and a testing framework ALPACA for performance and correctness concerns. Some research is about curricular experience with parallel computational thinking [6]. Especially, Lori Pollock et al. [7] utilize cooperative learning to meet the practical challenges of teaching parallel programming as well as to provide a more real world context of parallel programming.

The main problems we considered are as follows. (1) The syllabus, teaching contents and evaluation methods should keep pace with the development of HPC. How could we integrate our advanced scientific research results into our courses as classic cases? How could we combine teaching and research so that the teaching contents always keep up with the pace of HPC developments? (2) What could we do to activate the enthusiasm of the students with different programming levels? (3) Besides, how could we utilize the supercomputing center resources to build a teaching practical platform? There is no substitute for the practical available on a real parallel system [3]. The tests on a real supercomputer able to effectively utilize the underlying parallel hardware architecture.
In the past two years, we have used both of our advantages in HPC research and experimental platform to rebuild the first parallel computing series course: parallel programming. In details, we have done the following efforts: (i) We used national supercomputing center resources to build teaching practical platform. The purpose is to enable students to code on a real supercomputer system in order to train students in their parallel programming skills and parallel thinking for solving problems. Students learned the structural characteristics of parallel computer systems and the typical challenges faced by large-scale parallel scientific applications. And they greatly improved their practical programming ability. (ii) And furthermore, we ensured that the teaching practical topics were closely related to the latest HPC researches. We attracted HPC research group advisors or PhD students into our course, and formed a practice-centered curriculum model. Students obtained parallel thinking exercise from this curriculum model. (iii) By the demonstration and individual concern, as well as activity degree ranking mechanism, we inspired students’ research interest in HPC and related fields. By the two semesters’ practice, the effectiveness of curriculum reform won the praise of students and teachers.

The main contributions of this article are as follows. (i) We present the curriculum contents and methods as the center of improving parallel programming ability and parallel thinking. (ii) We present our efforts on the practice-centered curriculum model followed by practical experiences. (iii) We illustrate how we did the demonstration and individual concern, and how we did the activity degree of the ranking system of each individual student.

The structure of this article is as follows. Section 2 presents parallel programming curriculum goal and principle. Section 3 discusses practice-centered curriculum model and practical experiences. Section 4 gives some quantitative findings and Section 5, summarizes our findings and discusses our future work.

2 GOAL AND PRINCIPLE OF OUR PARALLEL PROGRAMMING COURSE

2.1 Goal of Our Course

The goal of this course is: (i) By coding directly on a real supercomputer in the national supercomputing center, students grasp parallel programming skills, as well as learning about parallel computer architecture; (ii) By integrating HPC related research topics into our project practice, students obtain parallel thinking exercise, which focuses on parallel programming ideas and basic parallel problem solving methods, as well as learning about representative challenges faced by parallel applications.

Figure 1 uses two big circles to outline the general framework we considered for parallel programming ability training and parallel thinking training. The inner cyan circle describes the ability training from the contents that we are concerned about. In terms of the teaching content for parallel programming ability improvement, the following things are necessary. They are commonly used parallel programming API (MPI, OpenMP, CUDA); parallel code debugging; parallel architecture oriented advanced programming skills; also project programming training. The outer yellow circle lists the ability training from the teaching methods what we care about, which refers primarily to the environment and condition to be provided. As far as parallel programming ability training, we provided the following three supports: real parallel computer systems, real parallel problems and incentive mechanism. Similarly, Two-layer circle on the right gives the teaching content and method about parallel thinking exercise. The content includes parallel programming idea, basic parallel problem solving methods and parallel architecture. Teaching method refers to project topics and the guide from HPC research group graduate advisors. The above two ability trainings cannot be completely independent. Some parts do overlap.

2.2 Principle for Our Course

The principle of parallel programming course is: lecture content and practice count for 50% of the course, respectively. Teachers are co-chaired by an instructor and several graduate advisors. The instructor is in charge of lecture course and graduate advisors instruct project practice. Practice-centered curriculum model adopts group discussion model aided by incentive mechanism. Student grades are mainly based on homework assignment, midterm project presentation, final project presentation and project experimental report.

(1) Basic curriculum design

The parallel programming course is the first course related to parallel computing, which is mainly for the first CS major or non-major grade students. This course is with both 36 hours of lecture and 36 hours of laboratory. The 36-hour lecture course includes: parallel architecture; distributed-memory programming MPI, shared-memory programming OpenMP and GPU programming CUDA; basic concepts related to parallel programming; parallel applications. And the 36-hour laboratory course mainly refers to project practice.

The question we are concerned most is: Q1: For the students who have higher parallel programming ability, do they definitely have stronger parallel thinking? What is the relationship between parallel thinking exercise and parallel programming ability?

Different from our previous parallel programming course, we provide an experimental platform supported by a subsystem of the national supercomputing center. Our students can login to the supercomputer subsystem via SSH and develop their parallel
applications using 36 computer nodes. Each compute node consists of two six-core processors.

It is necessary to pay the bill to use the supercomputing center resources. Students need to learn actual problems related to supercomputers although only supercomputing center users need to care about these problems. The advantage is that coding on a real supercomputer will motivate students to be more interested in parallel computer architecture and supercomputers. A teaching assistant is responsible for the answers to the questions, such as account numbers, compute node assignments, loading jobs and so on. User manual is ready for students in advance. For the characteristics of our supercomputer, we will refer to some special optimization questions and solutions.

Q2: Will the complex of real supercomputer systems bring more troubles or advantages to our course? What is the biggest advantage from coding on a real supercomputer?

(2) Practice-centered curriculum model

Project practice adopts group discussion model. 32 students are evenly divided into ten groups according to their research topics or directions. Each group performs a project topics including topic choice, project design, group discussion, mid-term presentation, final presentation, report writing and so on.

In order to integrate the latest HPC research contents into our course, we invite several graduate advisors to our course as project instructors, in charge of each phase of projects. Previously, graduate advisors and course instructors were completely isolated. This caused many problems and some complaints and misunderstandings appeared. Some graduate advisors thought some students didn’t have enough programming ability after studying lessons, which should be blamed on the curriculum. On the other hand, some course instructors thought the actual situations were too complex to solve immediately for a novice even if he had studied very hard in class. The instructor think the training is necessary. But research group advisors don’t allow enough practical time for the students to have hands-on time for work just learned. One reason of this phenomenon comes from different demands from graduate advisors and instructors. Graduate advisors hope students have enough ability to solve actual cases, not emphasizing the concepts (actually the concepts are key). While, instructors emphasize more on how to understand correct key concepts. They need to practice ability training because it can be pretty vague without real experimental platforms and real problems. The best solution is to eliminate the chasm between graduate advisors and instructors and to build a bridge between them. One way is to develop curriculum together.

Table 1: Distributions of Research Topics for a Master’s Degree

<table>
<thead>
<tr>
<th>Research Topics for a Master’s Degree</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>data analysis, big data</td>
<td>9%</td>
</tr>
<tr>
<td>deep learning, image recognition</td>
<td>31%</td>
</tr>
<tr>
<td>characteristics extraction, object detection</td>
<td>9%</td>
</tr>
<tr>
<td>neural network</td>
<td>16%</td>
</tr>
<tr>
<td>biological information</td>
<td></td>
</tr>
<tr>
<td>fluid dynamics, structure dynamics</td>
<td>19%</td>
</tr>
<tr>
<td>application simulation, numerical</td>
<td></td>
</tr>
<tr>
<td>weather prediction</td>
<td></td>
</tr>
<tr>
<td>architecture, robot OS, file system</td>
<td>25%</td>
</tr>
<tr>
<td>signal processing, human-computer</td>
<td></td>
</tr>
<tr>
<td>interaction, high-precision</td>
<td></td>
</tr>
</tbody>
</table>

We attract graduate advisors to join the course construction. Instructors and graduator advisors are in charge of lecture course

and practice together. The difficulty is how to attract advisors into courses, how to balance research topics and project topics, how to consider students’ interest when deciding project topics. A major reason to attract advisors is that we allow taking one part of research topics as our project topics, which makes our project practice is helpful to study a topic. Besides, most students taking part in the project topic are the members of this advisor’s research group, or those students with great interests in this topic choose this project, which is also helpful to the progress of the research topic. In this situation, students’ programming ability training and research topics are unified. After the projects belong to one part of the course, we need to tailor the content to a specific range, which is decided by group students and advisors. In fall 2015 semester, 32 students’ research topics include image recognition, deep learning, neural network, big data, biological information, fluid dynamics and so on. Table 1 lists more details about this. In the later context, Table 2 shows the statistics on relevance between project topics and research topics for a master’s degree. Here the question we care about is:

Q3: If the relevance between the project topic and research topic for a master degree is not very high, will the enthusiasm of the students be affected? Will the project practice be helpful to these research topics for a master’s degree?

Q4: In such group discussion models, how big is the difference of students? programming and thinking level in the same group? Will such kind of group discussion model help the programming ability improvement? Wouldn’t there be the students who fully depend on others in projects?

Q5: What is the role of presentation and writing project report for programming ability training during performing projects?

2.3 Incentive Mechanism

Most students would not like to discuss (maybe it is a popular phenomenon). Whatever in class or out of class, high-ability students always prefer to find answers by themselves, and those with poor academic performance always evade the discussion to avoid being focused by instructors. In this case, the instructor could not get an immediate feedback, which greatly influences the effect of the curriculum. Therefore, effective incentive mechanism is necessary.

Q6: Is it true that the more active the students are in cooperative learning, the higher score they will gain? Can the level of activity be inspired, which can be influenced by the atmosphere? Do those students with the worst activity definitely have the lowest scores?

3 PRACTICE-CENTERED CURRICULUM MODEL

3.1 Project Organization and Assessment Method

In each project group, one advisor is in charge of pushing and instructing the whole project, such as the topic selection, scheme preparation, weekly progress check and assessment, and the midterm/final presentation review. The instructor is in charge of communicating with the advisors, checking the weekly progress, arranging the mid-term/final presentations, reviewing the experimental reports and so on.

(1) Five stages for project
Project experiences the following five stages: definition, discovery, construct, refine and reflect. We evaluate a project according to the completion status of the above five stages and these three items: progress reports, self-evaluation and participation. So we have a detailed project evaluation rubric, which mainly refers to Example 1: Capstone Project in Design of CMU Grading and Performance Rubrics [1]. Our project evaluation rubric divides each item into three levels of achievement: sophisticated, competent and not yet competent. So this evaluation rubric can help students better understand what is expected of them through each stage of the project (More details see Appendix A). Figure 2 shows the distribution for all items in each stage. The rubric addresses the student’s work products, their presentation skills and their abilities to work. The content for each item sees Appendix A. All the students got this evaluation rubric before class starts.

**Stage 1: Definition**

Definition stage accounts for 2.5 points, which includes teambuilding and clarity of direction. Teambuilding demands team has energy and enthusiasm and each member has a clear role. Clarity of direction means clear hypothesis and a good plan draft for research. In terms of the level difference, the scores are divided into three grades. Most project topics are chosen guided by graduate advisors, and three other aspects.

**Stage 2: Discovery**

Discovery stage accounts for 15 points, which includes teambuilding and clarity of direction. Teambuilding demands team has energy and enthusiasm and each member has a clear role. Clarity of direction means clear hypothesis and a good plan draft for research. In terms of the level difference, the scores are divided into three grades. Most project topics are chosen guided by graduate advisors, and three other aspects.

**Stage 3: Construct**

Construct stage accounts for 30 points, which includes two parts: i) building a deep and logical connection between research and concept (10 points); ii) building experimental platform and determining experimental methods (20 points). Provide a detailed experimental step followed by experiment evaluation methods. According to the statistics, the former part has only 7 points averagely, and the latter part has 13 points averagely.

**Stage 4: Refine**

Refine stage accounts for 15 points including testing and refinement (8 points) and quality and level of completion (7 points). Testing and refinement demands team to use systematic testing to validate or drive refinement. The former part has 6 points and the latter one has 5 points in average. In this stage, each group performed the weekly discussion. During this discussion, they check the last week’s work and then decide the next week’s work. The advisor is in charge of checking all the work. All the records for all the groups are collected to the instructor. After the mid-term presentation, refinement focuses on all the questions by specialists and how to solve these problems and get improvement.

**Table 2: Research Topics for a Master’s Degree in Terms of Projects**

<table>
<thead>
<tr>
<th>Project Topics</th>
<th>Advisors for Projects</th>
<th>Research Topics for a Master’s Degree</th>
<th>Relevancy Between Projects and Research Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variation Oriented Parallel Text Mining of Biomedical Literature</td>
<td>Graduate advisors</td>
<td>scalable spectral clustering, distributed deep learning, medical image processing</td>
<td>30%, 10%, 20%</td>
</tr>
<tr>
<td>Genes Oriented Parallel Text Mining of Biomedical Literature</td>
<td>Graduate advisors</td>
<td>biological information, CFD simulation, text mining of biomedical literature</td>
<td>40%, 50%, 90%</td>
</tr>
<tr>
<td>CNN Convolution Parallel Computing</td>
<td>PhD student</td>
<td>object detection, speech recognition, dense matter structure and dynamics</td>
<td>10%, 80%, 0%</td>
</tr>
<tr>
<td>SAT Solution Parallel Speedup Technology</td>
<td>Graduate advisors</td>
<td>neural network based fuzzy technology, satisfiability problem based on many-core architecture, remote sensing image recognition</td>
<td>10%, 90%, 0%</td>
</tr>
<tr>
<td>MPI OpenMP Hybrid Parallel Computing for Parallel Spectral Transform Shallow Water Equation Model</td>
<td>Graduate advisors</td>
<td>feature extraction and classification about underwater acoustic based on deep learning methods, data assimilation, numerical weather prediction and marine environmental simulation, underwater acoustic signal processing</td>
<td>10%, 5%, 20%, 0%</td>
</tr>
<tr>
<td>Load Balance Optimization for CFD-MD Coupling Computing</td>
<td>Graduate advisors</td>
<td>CFD simulation, robot operation system</td>
<td>90%, 20%, 0%</td>
</tr>
<tr>
<td>Biological Effect Assessment based on Cell Reaction Big Data</td>
<td>Graduate advisors + PhD Student</td>
<td>biomedical big data, biological sequence alignment, Biological Magnetic Detection in Human - Computer Interaction</td>
<td>80%, 50%, 60%</td>
</tr>
<tr>
<td>3D Simulations of Human Ventricular Tissues</td>
<td>PhD student</td>
<td>network processor architecture, computer architecture, SSD distributed file system</td>
<td>0%, 0%, 5%</td>
</tr>
<tr>
<td>DPM Based Radiotherapy Dosage Calculation Parallelization</td>
<td>Graduate advisors</td>
<td>hemodynamics, high-precision computing, deep neural network model, data-stream based SVN techniques</td>
<td>40%, 30%, 0%, 30%</td>
</tr>
<tr>
<td>FEAST Based Un Paralleled Tolerance Optimization</td>
<td>Graduate advisors</td>
<td>image characteristics extraction by Spark, biological computing, AI</td>
<td>60%, 20%, 0%, 60%</td>
</tr>
</tbody>
</table>

**Stage 5: Reflect**

...into three grades. According to the statistics, the spectrum of information gathered is not broad enough, only 4 points averagely.
When he increased the scale, he found the memory size is not big which motivated him to find the actual answer for this question.

We found the difference between students’ interests and abilities is big. We paid an individual attention to those new discovered problems and led those students deeper into more extending problems. It could inspire those students’ interests in research and have a good influence on other students.

As homework give me an example. Write an MPI program that uses a Monte Carlo method to estimate PI. Student Guo was interested in the question: how a large scale can meet our demand for precision. At the first time, he drew a conclusion like that: when the scale is increased by 100 times, the precision of PI should be increased by 1 digit number. But the real test told him it is not true, which motivated him to find the actual answer for this question.

We demand each project to submit a full document covering weekly discussion records, weekly execution records, project definition stage document, mid-term presentation report, final presentation report, full experimental report, source codes, and so on.

(2) Project Presentation

It is very necessary to organize one mid-term project progress presentation for finding and correcting problems. Before presentation, each group submitted one report and full record documents. On the presentation day, one student reported the whole group work and every other member needed to report the individual work, including task assignment, completion status, questions and so on. The time for each group’s presentation and answer was within 20 minutes. Reviewers consisted of the instructor, group advisors and other invited experts. Each reviewer needed to ask questions and gave a remark. The mid-term presentation cost us a two-hour class time. Appendix B shows a review summary for a project group.

By the mid-term presentation, many technique problems are discovered. We summarized all the comments and found that the technical issues occupy 75%, the issues caused by wrong principles occupy 10%, expression issues occupy 8%, slow progress issues occupy 5% and other issues occupy 2%. From the review scores, the deviation between ten groups is about 8%. In each group, the score deviation between members is about 10%.

3.2 Individual Attention and Incentive Mechanism

3.2.1 Individual Attention and Demonstration Effect. During our practice class, students finish project coding and other coding homework. We found the difference between students’ interests and abilities is big. We paid an individual attention to those new discovered problems and led those students deeper into more extending problems. It could inspire those students’ interests in research and have a good influence on other students.

MPI communication routines. For example, he found that when the number of divided memory sections is increased, the time for Scatter operations will be reduced. Student Guo presented a report in class, which made several other students interested in this problem. Then they reworded the list of all the questions, and then redesigned the experiments. After many tests, they have a deeper understanding of parallel architecture, distributed memory system and global communications.

This practice proves that we really can extend a simple exercise to a series of tests and trials so that students can get better understanding for some key concepts in parallel computing. The key point is that it actually comes from the new discovery of students. For question Q2, the test on a real supercomputer makes Student Guo find problems and motivate his and other students’ interests. So the complex of real supercomputer systems does bring more advantages to our course. The biggest advantage from coding on a real supercomputer is that it can provide the students with the opportunities to find the unknown things.

3.2.2 Incentive Mechanism by Active Ranking. It is very important to encourage the students to participate in-class and out of class discussions. It helps to solve the problems in a short time, give an immediate feedback and stimulate the classroom’s atmosphere. We set a point accumulation method to inspire the enthusiasm of students’ discussions. All kinds of discussions are encouraged, such as classroom questioning, practice questioning, Wechat group discussion, online discussion forums and so on. Each effective question is marked as one point, and one answer is marked as 0.5 point. A teaching assistant counted one active ranking each week and published the ranking result every two or three weeks. There are nine weeks in total.

![Figure 3: The relationship between the level of activity and course scores.](image)

![Figure 4: The relationship between the level of activity and course scores.](image)
to the level of activity. We obtained this deviation by subtracting the average level of activity from the average course score. For Q6 (1), it is right from the general sense. But there exists exceptions. For example, No. 7 student in Figure 3, who has a high level of activity (82.5 points), only has 74 scores. Further, we find 25% of the questions can be seen as a spam message and 25% of the questions are about basic concepts and also 25% of the questions are about GDB skills and other 25% are answers to other questions. For Q6 (2), the level of activity will really be affected by others from Figure 4 but this influence is very limited. The area where the green area is bigger than the red and the blue areas indicates the increasing level of activity in the last three weeks. But, there also exists the area where green area is smaller and smaller and even disappears. This shows some students even took no discussions for a period of time. In fact, in the last three weeks, there are about 34% students who took not any discussions. Some causes are project progress, report submission deadline, other course exams. For Q6 (3), according to our statistics, those students who never took discussions have lower scores than the average score. In those students, one student score is nearly the lowest, and other students’ scores are 1.1% lower than the average score.

4 QUANTITATIVE FINDINGS

In July 2017, Totally 32 questionnaires were sent out, 31 were effectively received with effectively recovery of 97%. These students will get the master degree at the end of 2017. Through this questionnaire, we can learn whether the first parallel programming did help their master research topics or not. The contents include: How is the relevance between this course and their master research topics? How much help do they get from this course lectures, especially for their research topics? How much help do they have from this project practice for their research topics? How were they feeling about the incentive mechanism by active ranking? and so on. Combining other source scores and statistics, we get the following quantitative findings.

4.1 Course Lecture Part

According to the questionnaire feedback statistics, 28% of the students think the lecture content really greatly helps them, 66% of the students think it helps them some. 3% of the students think this help is unnecessary. Students also listed several items they think most helpful, such as parallel programming idea and problem solving methods (41% of the students agree with that), parallel code debug (38%), scalability and speedup related concepts (34%), parallel architecture (25%), MPI global communication (22%), MPI-OpenMP-CUDA (13%), others (3%). From the above feedback, most students think methodology and ideas can help them the most. They have this feeling especially when they studied the graduate research topics, which is exactly one of our course goals. Secondly, students consider not only programming skills are important, but also debug skills are more important, which can improve the coding efficiency. Lastly, several key concepts for large-scale parallel computing, such as scalability, speedup, really help them to understand and solve their problems in their later research. In the previous course development, these contents are usually taught by lectures instead of practice. Now, more exercises and practices deepen them to understand these concepts. In all, more than one-third of the students think the lecture part did help them a lot.

4.2 The Most Helpful Practical Contents to Their Master Research Topics

For the project practice, the contents that are most helpful to their master research topics are listed in Table 3. In terms of the statistics, about 38% of the students think several parallel programming API gave them a lot of help in their master research topics, which coincides with our major teaching goal. Then, 28% of the students think parallel thinking training and parallel problem solving methods, have contribution to their research. Besides, some students think parallel architecture is very helpful.

Table 3: The Most Helpful Practical Contents to Their Master Research Topics

<table>
<thead>
<tr>
<th>the Most Helpful Contents</th>
<th>Proportion of Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI, OpenMP, CUDA</td>
<td>38%</td>
</tr>
<tr>
<td>parallel thinking training, parallel problem solving methods</td>
<td>28%</td>
</tr>
<tr>
<td>parallel architecture</td>
<td>3%</td>
</tr>
</tbody>
</table>

4.3 Feedback from Projects

22% of the students think project practice has a big help to their research. 59% of the students think it has certain help. 13% of the students think it has not big help. Other 3% think it has no help. Students also gave several most helpful project practical contents: problem analysis (47% of the students agree with it), coding (44%), presentation (31%), teamwork (28%), literature reading (25%), report writing (16%), experiment design and result analysis (3%). From the above statistics, most students think project problem analysis is most beneficial. The second is coding training. The students who have higher parallel programming ability are not sure to have stronger parallel thinking. The relationship between parallel thinking exercise and parallel programming ability is that parallel programming is a technique, a tool while parallel thinking belongs to one computational thinking and is a fundamental method (Q1).

One student specially mentioned, although this course had no big relevance to his master research topics, he got good programming training through this course and his teamwork won the third prize during 2016 parallel application challenge competition. This is a good evaluation for our course. We spent two-hour class to have the presentation. 31% of the students think the two presentations gave them a big help. This shows the role of presentation (Q5).

For question Q4, there really exists difference in one project group, but group model actually helps a lot for students’ programming ability improvement by comparing mid-term and final presentation review scores. Few students fully depend on others in the same project group.

5 CONCLUSIONS AND FUTURE WORK

This article summarizes our practice for the first parallel programming in our university. We list some problems we are facing and want to solve. Then, we propose our goal for this course followed by principles of the curriculum. Particularly, we present our practice-centered curriculum model and practical experiences followed by quantitative findings.
REFERENCES


