Olympiads in Informatics, 2013, Vol. 7, 180–185

Arbiter: the Evaluation Tool in the Contests of the China NOI

Qiyang ZHAO[1], Fan WANG[1], Baolin YIN[1], Hui SUN[2]

[1]*School of Computer Science and Technology, Beihang University 100191 Beijing, China*

[2]*Information School, Renmin University of China*

e-mail: zhaoqiyang@nlsde.buaa.edu.cn, wangfan@nlsde.buaa.edu.cn, yin@nlsde.buaa.edu.cn, sun_h@ruc.edu.cn

Abstract. In this paper, we introduce the evaluation tool, Arbiter, adopted in all contests of the China NOI(National Olympiad in Informatics). Due to the diversity of contest environments, the Arbiter family consists of three types of evaluating tools, ACC, AOC and ALS, in order to fit the camp contests, online contests, and large scale contests respectively. These tools are distinguishable both in technical details and user interfaces. Since 2002, Arbiter has been widely used in all NOI contests as the unique official evaluation tool.

Key words: grader, China NOI, evaluation.

1. Introduction

The contests of the China National Olympiad in Informatics (NOI) include the National Olympiad in Informatics in Provinces (NOIP), NOI Camps, and NOI Online contests (China National Olympiad in Informatics – about). These contests differ both in scale and type. In order to provide comprehensive support, a group of program grading and evaluation tools, the Arbiter family, was developed to fit the different contest environments. These tools run on most popular Linux OSs, including *Ubuntu*, *Redhat*, *Debian*, etc.

The Arbiter family, named after its role, has supported all NOI contests since 2002, from large scale contests of nearly 100,000 junior contestants in 31 provinces to camp contests involving 400 top contestants competing in a single hall (China National Olympiad in Informatics).

2. Arbiter for Camp Contests (ACC)

The *ACC* works in the *Client-Server* mode on a local network as shown in Fig. 1. The entire system comprises a control server and all the machines that the contestants use in the competition along with their submitted programs on those machines. The main parts of the ACC run on the control server, while the contestants' programs are evaluated

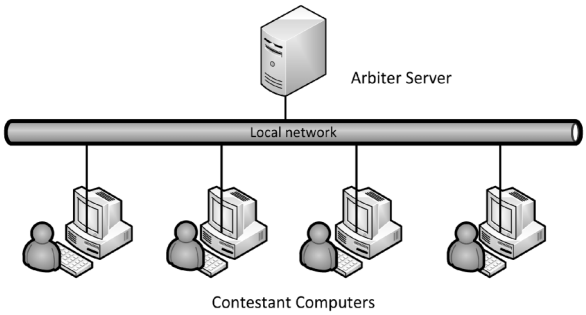


Fig. 1. Arbiter for camp contests.

on their local computers in a highly parallel way. In general, all evaluation tasks can be finished in couple of minutes, and no extra machines are needed.

The control server directs and monitors contest machines in all stages: transferring data, evaluating contestants, and so on. All Arbiter clients work on contestant machines in the daemon mode, receiving and executing commands from the control server, evaluating the contestant answers independently, and filtering all suspicious network packages sent to/from host machines. Arbiter servers and clients communicate with each other by UDP broadcasting and TCP to obtain highest efficiency and reliability as possible.

The evaluation of a camp contest is conducted in four stages: *preparation*, *contesting*, *evaluation*, and *appealing*.

(1) In the **preparation stage**, the server randomly assigns all contestants to the computers. The contest settings and contestant accounts are set, and the contest data, including problem descriptions and sample data, are sent to all contest machines.
(2) In the **contest stage**, all the Arbiter clients monitor contestant behaviors and network packages. Any exceptional actions are reported to the control server to notify contest supervisors so they can deal with it in a timely manner.
(3) In the **evaluation stage**, the server commands all clients to evaluate the contestants' answers on their own machines. Since the evaluation tasks are performed concurrently, all the work of this stage can be finished in minutes. The evaluation scores are sent to the control server, and the results can be checked and forms can be generated for the administration use. After approval, the evaluation results are sent to the corresponding machines of the contestants.
(4) In the **appeal stage**, contestants read the evaluation reports on their own machine, and create appeals for re-evaluation if they see a problem. The ACC is capable of documenting all evaluation histories and solidly supporting the judgment on whether the appeals are reasonable or not.

The control server adaptively chooses suitable schemes for efficiently transferring the contest data. All schemes are based on UDP broadcasting or TCP-based point to point, which is similar to the popular TCP-based BT mode. Data sized of hundreds of MB (being zipped) will be distributed to $400+$ contestant machines in couple of minutes. This is very efficient and meets requirements of all the contests, even one with extraordinarily large testing data.

3. Arbiter for Online Contests (AOC)

In online contests, all contestants are required to submit their answer files to a remote server. The evaluation tasks are automatically assigned to and graded by several evaluating computers, as shown in Fig. 2.

All online contests run in the *Web-Server* mode via Internet instead of a local network. The Arbiter tools are installed only on the contest server, and there are no special requirements on contestant machines (except a web browser, of course). The contestants are required to register on the remote server, read/download the tasks, and upload their answers. Behind the contest server, a cluster of machines evaluates all contestant answers and reports the results to the server. Adding additional machines to or removing existing machines from the cluster is simple. The contestants can get their scores/ranks immediately or in 1–2 days after the check and approval conducted by the administrator, according to the configuration in advance.

An online contest proceeds in four stages: *registration*, *contesting*, *evaluation*, and *appealing*.

(1) In the **registration stage**, contestants register themselves on the Arbiter server and get their accounts and passwords.
(2) In the **contest stage**, contestants log into the Arbiter server and read the tasks on the contest web pages. If configured, the contestants can run their answers on the sample data provided and obtain the feedback from the server.
(3) In the **evaluation stage**, the Arbiter server assigns contestants' answers to the evaluating machines and gets the evaluation results from them. Contestants can read their scores and evaluation specifications on contest web pages.
(4) In the **appealing stage**, contestants submit their appeals to the Arbiter server and wait for further processing. After proper processing, the final scores will be published on official NOI web pages.

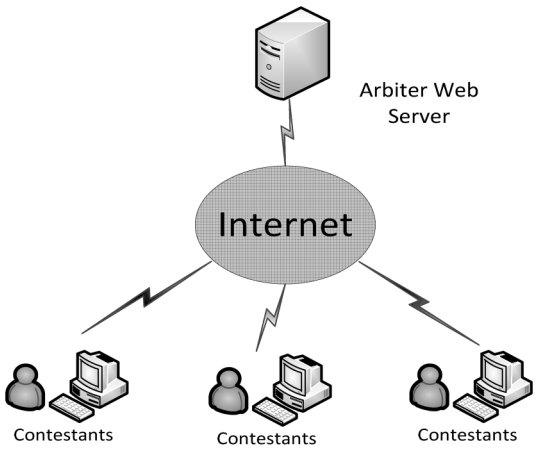


Fig. 2. Arbiter for online contest on Internet.

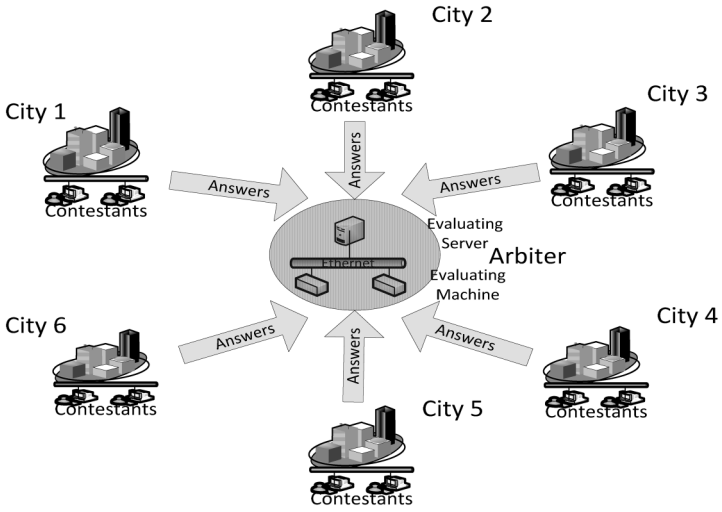


Fig. 3. Arbiter for large scale contests in different cities.

4. Arbiter for Large Scale Contests (ALS)

The NOIP contests are held simultaneously in 31 provinces for about 100,000 contestants. In order to keep the unique standard of evaluation over the provinces, the answers of all contestants are gathered together via email and evaluated in an off-line mode by the ALS system, as shown in Fig. 3.

The ALS runs on an evaluation server in either parallel mode or distributed mode, or both, according to its hardware environment it. At the initiating stage, the ALS tests for the CPU architecture and the network environment. For the multi-core CPUs on evaluating machines, the ALS will assign an evaluating process for each core. If the evaluation server is connected to a cluster of evaluating machines on which the Arbiter evaluators are installed, the ALS will send them the proper number of evaluation tasks according to their capacities, as shown in Fig. 4. In this way, the ALS makes full use of the hardware environment so that the evaluation tasks can be done in high speed.

5. Features of the Arbiter Family

In general, the Arbiter tool family exhibits the following features:

(1) *Flexibility in Usage*
It affords great convenience to contestant administrators in contest administration in all stages. The administrators can easily configure and control contests as required, such as changing the number and the data of the test data, setting the weight of each test case, publishing the results, and so on. The Arbiter server is developed with popular languages and tools, such as C, PHP, JSP, and QT Designer, which ensures compact views, convenient operating interfaces, and comprehensive shortcuts. When designing and developing

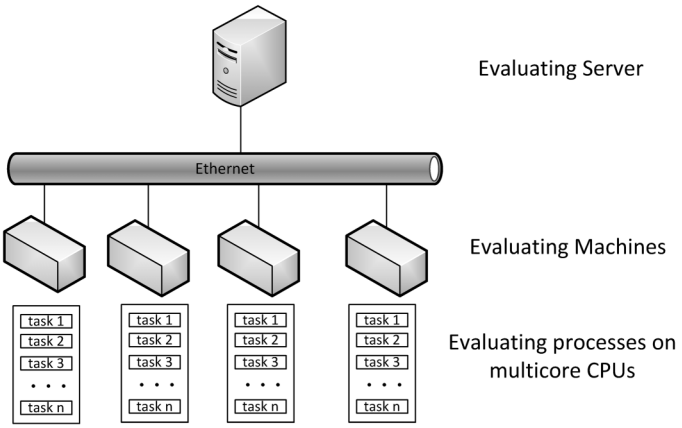


Fig. 4. An evaluating cluster of ALS.

Arbiter tools, a fundamental principle is maintained firmly: there should be fewer than four operations to resolve any complicated administrative affair. This simplifies the operating flows and makes the Arbiter easy to use.

(2) *Stability and Efficiency in Evaluation*
In our experiments, Arbiter is very stable with an error rate less than 0.1%. All contestants' answers are evaluated on contestants' machines or evaluating clusters concurrently, during which Arbiter records and checks all resources used, including CPU time and memory usage. The system will stop the evaluation processes if the program being evaluated runs 100ms more than the specified time. Therefore, little time is wasted if the constant's program runs beyond the time limits.

(3) *Easy to Migrate*
Almost all Arbiter tools are developed with C programming language, while the UI pages are written with QT libraries. Therefore, the Arbiter family is easily migrated to other operating systems with little cost. Furthermore, Arbiter is not designed for evaluating programs in a specified language. It is independent to the compliers of the target programs being evaluated. Therefore, programs in any programming languages can be evaluated, provided the corresponding compilers or interpreters can be found. It is an advantage for the contest organizers when attempting to provide new languages in the contests.

(4) *Standard Interfaces for Data Exchange*
In order to accommodate popular word-processing software such as Open Office, MS Office and so on, Arbiter restores data in standard formats, such as CSV, PostScript, etc. Therefore all configurations, contestant information, and score/rank lists can be easily imported from external data sources or be exported for further use.

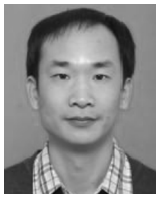
(5) *Secrecy in Data Storage*
Arbiter protects all critical data by encryption and access controls. The contestants are unable to access the data files directly, and all data operations must be performed using Arbiter tools, in accordance to the contest regulations.

6. Conclusions

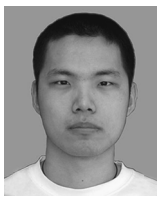
Up to now, the Arbiter tools have evaluated about 700,000 contestants, 5,000,000 problem answers, on about 60,000,000 test cases. According to official statistics published by the NOI Scientific Committee, the error rates of these evaluating tasks are about 0.05%–0.25%. It shows that Arbiter family is stable and efficient, especially in large scale contests. Therefore, the Arbiter family has become the official evaluating system for all the NOI contests since NOI 2002 Summer Camp.

References

China National Olympiad in Informatics. `http://www.noi.cn/about/summary/`
China National Olympiad in Informatics. `http://www.noi.cn/noi2009-noip2009/`

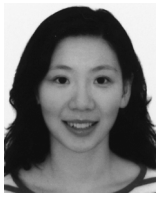


Q. Zhao, dr., is associated with the School of Computer Science and Engineering (SCSE), Beihang University (BUAA) of China. His major interests are in computer vision and image processing. He is a member of NOI science committee since 2009.



F. Wang is now pursuing his PhD degree in the School of Computer Science and Engineering (SCSE), Beihang University (BUAA) of China. There he received his master degree in 2011. His major interest is in computer vision.

B. Yin is a professor at the School of Computer Science and Technology, Beihang University of China. He received his PhD degree in computer science from the Department of Artificial Intelligence, the University of Edinburgh in 1984. He has served as the vice chairman of Scientific Committee of National Olympiad in Informatics since 1999.



H. Sun, dr., is associated with Information School, Renmin University of China. She received her PhD degree from Tsinghua University in 2005. Her research interests are in data mining. Since 2005, dr. Sun is a member of NOI Science Committee and she will be deputy leader of China Team in IOI 2013.