

BOCA: um sistema de apoio a competições de programação

Cassio P. de Campos¹, Carlos E. Ferreira²

¹ Pontifícia Universidade Católica de São Paulo
Centro das Ciências Exatas e Tecnologia
Rua Marquês de Paranaguá, 111
Consolação - São Paulo - SP - Brasil

² Universidade de São Paulo
Instituto de Matemática e Estatística
Rua do Matão, 1010
Cidade Universitária - São Paulo - SP - Brasil

cassio@pucsp.br, cef@ime.usp.br

Abstract. *In this article we describe BOCA, a software for supporting programming contests developed to be used in the Maratona de Programação of the Brazilian Computing Society. The system can also be used for supporting disciplines that make evaluation of student assignments during the classes.*

Resumo. *Neste artigo apresentamos o BOCA, um sistema de apoio a competições de programação desenvolvido para ser usado na Maratona de Programação da Sociedade Brasileira de Computação. O sistema pode ser usado também no apoio a disciplinas em que se faça uso de submissão e correção de trabalhos durante as aulas.*

1. Introdução

Competições de programação são uma excelente oportunidade para desenvolver nos alunos diversas habilidades que serão extremamente úteis no seu crescimento profissional. Nelas, os alunos aprendem de forma divertida conceitos importantes sobre estruturas de dados, algoritmos e desenvolvimento de software enquanto, ao mesmo tempo, aprendem a trabalhar em grupo e sob pressão. Dois exemplos de competições deste tipo são o Concurso de Programação da ACM (o ICPC: *ACM International Collegiate Programming Contest* [1]) e as Olimpíadas Internacionais de Informática (IOI) [5].

O ICPC da ACM destina-se a alunos de instituições de ensino superior e em 2004 realizou sua 28^a final mundial em Praga, República Tcheca. As finais mundiais da 16^a IOI serão em Atenas, Grécia.

O sucesso destas competições tem sido extraordinário. No ano de 2003 competiram em regionais para o ICPC mais de 3000 times (cerca de 1400 instituições de ensino superior) de 75 países de todo o mundo. Destes, apenas 72 times têm o privilégio de competir nas finais mundiais pelo título de melhor equipe de programação do planeta. No último ano este título pertenceu à Universidade de Varsóvia, Polônia, e após as últimas finais o título passou à St. Petersburg University of Information Technology, Mechanics & Optics, Rússia.

No Brasil temos verificado interesse semelhante nestas competições. A Olimpíada Brasileira de Informática [7] realizou em 2003 sua quinta edição e contou com a participação de mais de 1300 alunos na seleção dos 4 brasileiros que compuseram o time

nas finais mundiais do IOI, nos Estados Unidos. A Olimpíada é realizada pela Sociedade Brasileira de Computação, sob coordenação do Prof. Ricardo de Oliveira Anido, do IC Unicamp, e com apoio do CNPq.

Desde 1996 vêm ocorrendo no Brasil competições de programação seletivas para as finais mundiais do ICPC. Naquele ano, a competição foi organizada pelos Profs. Claudionor Coelho e Ricardo Dahab (IC-Unicamp, que desde então tem sido o diretor da regional sulamericana da competição) e 13 times competiram em Belo Horizonte pelo direito de representar o país nas finais mundiais do concurso, que foi ganho pelo time do IC da Unicamp. O time do IC Unicamp disputou as finais mundiais de 1997 em San Jose, California.

No ano 2000, a competição (chamada no Brasil de Maratona de Programação [6]) passou a ser realizada pela Sociedade Brasileira de Computação e coordenada por Carlos E. Ferreira, do IME-USP. Na tabela abaixo mostramos a evolução da participação de times e escolas brasileiras na Maratona de Programação desde 1996.

Ano	Escolas	Times
1996	13	13
1997	18	18
1998	21	21
1999	22	26
2000	26	40
2001	39	68
2002	61	114
2003	60	119

Como pode ser verificado na tabela acima, a cada ano o interesse cresce, e a competição envolve mais e mais escolas e estudantes. Em muitas escolas há iniciativas de preparar equipes de alunos para a competição através de disciplinas optativas, onde os alunos aprendem técnicas de programação como divisão e conquista, algoritmos gulosos, programação dinâmica, etc. O aprendizado é baseado na solução de desafios que fizeram parte de competições anteriores. Neste sentido, podemos inclusive encontrar livros abordando diretamente esse assunto [12]. A idéia deste trabalho é colocar à disposição de alunos e professores o sistema de apoio que vem sendo utilizado na Maratona de Programação desde o ano de 2002 com bom desempenho. O sistema está à disposição dos interessados no site do BOCA [3].

Este trabalho está organizado da seguinte forma. Na próxima seção descrevemos o ambiente de programação disponível para os alunos na Maratona de Programação. Na seção 3 descrevemos as principais características do BOCA. O trabalho encerra com algumas conclusões.

2. Ambiente de Programação

Durante a realização de uma competição oficial, temos algo semelhante à realização de uma prova. Os times não podem ter acesso a nenhum material externo ao seu computador (não podem ter acesso à internet e nem aos arquivos dos outros times).

Na competição, cada time tem acesso a um único computador, onde devem estar presentes ferramentas de desenvolvimento de software, como compiladores, editores de texto, ambientes integrados de desenvolvimento, depuradores, etc.

É necessário que exista um mecanismo para a entrega dos programas em tempo real, já que a correção e resposta é feita durante a prova e a informação sobre o acerto ou erro deve ser passada assim que possível para o time.

Uma maneira de controlar o acesso dos times, disponibilizando apenas o que é necessário, é através de um sistema criado exclusivamente para tal finalidade. Assim nos últimos dois anos a Maratona de Programação vem usando uma distribuição Linux (denominada *Maratona Linux*) projetada para atender suas necessidades.

A distribuição *Maratona Linux* foi inicialmente criada com base na famosa distribuição *Red Hat Linux* [11], mas sua idéia não está presa à mesma.

O *Maratona Linux* foi projetada para ser uma distribuição de instalação muito fácil, e assim seu uso não necessitaria da presença de uma especialista em administração de sistemas Linux. Durante uma competição normalmente são utilizados os laboratórios onde outras atividades são realizadas antes e logo após o término do evento. No *Maratona Linux* apenas um computador é instalado "desde as raízes" (é aquele que será o servidor), perdendo-se o conteúdo do disco rígido. Nos computadores que são utilizados pelos times apenas um arquivo precisa ser copiado, sem a necessidade de apagar nada. Dessa forma, mesmo que no mesmo dia o laboratório precise ser utilizado por outros, basta reiniciar os computadores que o sistema antigo estará funcionando.

Durante a instalação do servidor, o sistema encarrega-se de criar as configurações para todos os demais computadores que serão usados na competição. Além de configurações dependentes do hardware das máquinas, são criadas regras para bloquear acessos ilícitos.

Todos os dados voláteis do computador de cada time (arquivos de configuração, arquivos de trabalho do time, etc) ficam armazenados no servidor. Isso possibilita também uma fácil troca de computador caso haja problemas. Somente o servidor não pode ser trocado de maneira trivial.

A instalação é feita de forma dirigida, onde o responsável precisa apenas responder simples questões. Ao final, já se tem um sistema pronto para iniciar uma competição, que é gerenciada pelo BOCA (incluído também no *Maratona Linux*).

3. Sistema BOCA

Run #	Time	Problem	Language	Answer
2	37	Problema 1	C	No - Compilation error
3	37	Problema 3	Java	Yes
4	83	Problema 1	C++	Not answered yet

To submit a program, just fill in the following fields:

Problem:

Language:

Source code:

Figura 1: Interface do time para enviar submissões.

Na competição, a correção dos programas enviados pelos times é feito de forma online, e o resultado deve ser transmitido ao time o mais breve possível. Temos aqui uma diferença fundamental com relação à comparação a uma prova tradicional, feita na seção anterior. Imagine que, durante uma prova tradicional, os alunos entregam suas questões separadamente assim que as terminam, e o professor deve prontamente corrigi-las, informar ao aluno se a mesma está certa ou errada, e caso esteja errada o aluno tem a possibilidade de tentar consertá-la e entregá-la novamente, ainda durante o tempo de prova, mas

sem saber especificadamente o que estava errado na questão. Esse é basicamente o mecanismo das competições de programação citadas. Assim, podemos dizer que a principal função do BOCA é executar esse serviço durante uma competição. Na figura 1 podemos ver a interface disponível para o aluno enviar suas soluções às questões propostas pelos juízes.

Continuando a analogia, as dúvidas que os alunos têm com relação à prova também podem ser enviadas pelo sistema, sem a necessidade de ficar chamando o professor. O professor pode responder à dúvida de forma individual ou coletiva, sempre através do sistema. Assim o BOCA internamente tem disponível um esquema similar a um chat/fórum de discussões, mas que funciona apenas de forma dirigida e controlada. Na figura 2 podemos ver a interface disponível para um juiz (ou professor) responder às perguntas feitas por alunos.

Use the following fields to answer the clarification:

Clarification Site: 2
Clarification Number: 1
Clarification Time: 86
Problem: Problema 1

Clarification:

Answer:

Answer all users in the site
Answer all users in all sites

Answer No response Cancel Clear

Figura 2: Interface do juiz para responder a perguntas feitas pelos times.

Desenvolvido com uma interface web, o BOCA é um sistema de entrega de exercícios, com autenticação, controle de tempo e disponibilização de resultados, tudo em tempo real. Toda a programação do sistema foi feita na linguagem PHP [8], e assim é portátil para todo sistema onde tal linguagem esteja disponível. Para armazenamento dos dados e controle de concorrência utilizamos o banco de dados relacional PostgreSQL [9]. Vale ressaltar neste ponto que o sistema é flexível para acomodar trocas de tecnologia. Ambas as tecnologias são livres e não trazem custos adicionais. É necessário ainda um servidor de páginas onde o sistema estará incubado. O Apache [2] (também livre) é uma boa escolha para tal.

BOCA Login

Name Login

Password

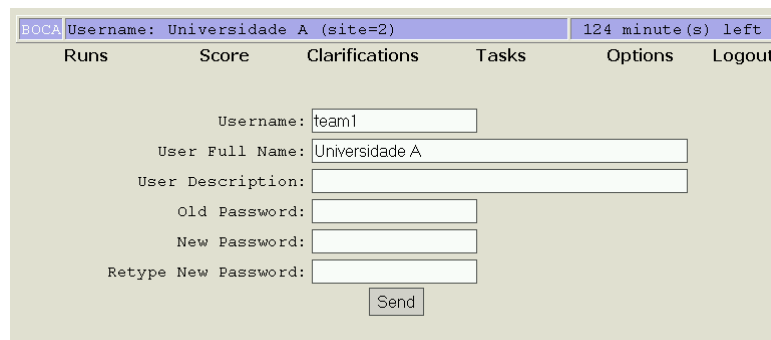
Figura 3: Interface para autenticação no sistema.

Podemos dividir o BOCA em cinco partes, de acordo com a especificidade de cada usuário: time, juiz, administrador, staff e placar. Cada uma dessas partes tem interface própria e diferenciada, de acordo com as necessidades e responsabilidades de

cada perfil. Note que não existem diferenças entre os computadores que são usados pelos juizes, administradores, times, etc, a não ser pelo fim que é dado em cada caso. Assim, o *Maratona Linux* pode ser usado em todos os computadores.

3.1. Time

Os times possuem uma interface simples e direta, onde é possível enviar os programas para correção, alterar informações pessoais, submeter dúvidas aos juizes e verificar o placar *online* da competição. Além disso, durante a competição, os times podem solicitar através do sistema a impressão de seus códigos-fonte e ajuda do pessoal de staff. O ambiente do time (descrito nesta seção) e a parte do placar (descrita na seção 3.5) são as interfaces mais simples do sistema.



The screenshot shows a web browser window with the BOCA interface. At the top, it displays 'BOCA Username: Universidade A (site=2)' and '124 minute(s) left'. Below this is a navigation menu with tabs for 'Runs', 'Score', 'Clarifications', 'Tasks', 'Options', and 'Logout'. The main content area contains a form for updating user information. The fields are: 'Username:' with the value 'team1', 'User Full Name:' with the value 'Universidade A', 'User Description:' (empty), 'Old Password:', 'New Password:', and 'Retype New Password:'. A 'Send' button is located at the bottom of the form.

Figura 4: Time alterando seus dados cadastrais e senha de acesso.

No momento de submeter seu arquivo-fonte (tipicamente chamado de uma *run*), o time pode escolher a linguagem que o programou e para qual problema ele é destinado. Através do ambiente de janelas do navegador, o time escolhe o arquivo do disco que deseja submeter e o envia para correção dos juizes. Na mesma janela de submissões, o time pode verificar o andamento das correções de submissões passadas feitas por ele, como mostrado na figura 1.

No item envio de perguntas (ou *clarifications*), o time pode escolher sobre qual questão da prova deseja fazer a pergunta (ou se a mesma é geral) e digitar diretamente a dúvida. Nesta mesma janela, o time acompanha as dúvidas feitas anteriormente por ele e também dúvidas de outras equipes que, por decisão dos juizes, foram escolhidas para serem divulgadas entre todos.

No item *score*, o time pode acompanhar em tempo real o placar da competição, podendo até mesmo usá-lo como decisão de que rumo tomar na continuação da sua prova. Equipes experientes costumam usar o placar como indicativo de qual a melhor ordem para atacar as próximas questões.

Resta ainda a janela de *tasks*, onde o time envia seus arquivos para impressão ou chama por assistência do pessoal de staff. Em ambos os casos o time é atendido diretamente no local onde está fazendo a prova, onde uma pessoa da organização leva-lhe a impressão ou oferece-lhe ajuda.

3.2. Juiz

A interface do juiz permite ao mesmo responder perguntas e dúvidas de forma individual ou coletiva, além de permitir ao juiz criar perguntas globais para serem respondidas a todos. O juiz tem a possibilidade de analisar e corrigir os programas enviados pelos

times, definindo no sistema qual a resposta que o time receberá para sua tentativa. As repostas possíveis podem ser configuradas para cada competição. Nas competições

citadas no início do texto, utilizamos respostas como **certo**, **saída errada**, **saída mal formatada**, **erro de compilação**, **erro em tempo de execução**, etc. Nada além desse tipo de informação é transmitido ao time, que deverá achar seu próprio erro no caso de uma resposta negativa. O juiz pode também acompanhar o placar *online* da competição no seu próprio computador ou alterar seus dados, como nome, filiação, senha, etc.

Figura 5: Juiz respondendo uma submissão de um time.

No item *runs*, o juiz acompanha quais as submissões foram feitas pelos times e que ainda não foram completamente processadas. Aparecem nesta janela as *runs* que estão sendo julgadas por outros juizes, pelo próprio, e ainda aquelas que não começaram a ser julgadas.

A opção *clarifications* é muito semelhante à opção *runs*, porém existe ainda a possibilidade do juiz incluir uma dúvida própria para ser respondida por ele mesmo ou por outro juiz. O objetivo deste tipo de procedimento é divulgar, de forma global, algum detalhe dos problemas e/ou enunciados que não ficou claro após a confecção do caderno de questões da prova.

Para verificar tudo que foi feito por ele, o juiz tem ainda à disposição o item *history*, onde pode acompanhar todas as respostas de perguntas passadas e todas as *runs* que foram julgadas por ele.

3.3. Administrador

Figura 6: Administrador tem controle sobre a configuração inicial da competição, como a inclusão de problemas.

A interface do administrador da competição é a mais sofisticada. Através dessa interface devem ser configuradas as informações sobre a competição, sobre as linguagens

de programação que serão permitidas e as questões da prova. É através dessa interface que é feita a inclusão de times, juízes, placares, staff e administradores, onde a prova é definida, iniciada e interrompida, e onde se pode também ter acesso aos registros de tudo que está ocorrendo no sistema. São permitidas alterações em diversos dados da competições (inclusive nas submissões, programas, perguntas, controle de acesso ao sistema).

The screenshot shows the OJ system administrator interface. At the top, there is a navigation menu with the following items: **Runs**, **Score**, **Clarifications**, **Users**, **Problems**, **Languages**, and **Answers**. The **Runs** item is selected, and its sub-menu is visible, containing: **Tasks**, **Site**, **Contest**, **Logs**, **Reports**, **Options**, and **Logout**. Below the menu, there is a table of submissions with the following data:

Run #	Site	User	Time	Problem	Language	Status	Judge (Site)	Answer
5	2	team2	98	Problema 1	C	judged	judge1 (2)	Yes
4	2	team1	83	Problema 1	C++	judging	judge1 (2)	Not answered yet
1	2	team1	39	Problema 1	C	deleted	admin1 (2)	Yes
3	2	team1	37	Problema 3	Java	judged	admin1 (2)	Yes
2	2	team1	37	Problema 1	C	judged	judge1 (2)	No - Compilation error

Figura 7: Administrador pode fazer o acompanhamento completo da competição, como a verificação das submissões dos times.

O usuário administrador possui diversas janelas diferentes para controlar o sistema. A seguir, descrevemos suas funções principais:

- **Runs:** nesta pasta estão apresentadas na tabela todas as submissões, com informações sobre elas. O administrador pode retirar uma submissão de um juiz, apagar uma submissão, julgá-la ou reabri-la após já ter sido corrigida. Clicando sobre uma submissão é aberta uma tela para julgamento da mesma (note que o administrador pode até julgar uma *run*, mas isso deve ser feito excepcionalmente para corrigir erros de julgamento).
- **Score:** Nesta pasta o administrador pode acompanhar o andamento do placar da competição.
- **Clarifications:** Esta opção é semelhante à *Runs*. O administrador pode apagar, reabrir, retirar de um juiz ou responder a uma dúvida. Ao clicar sobre o número de uma pergunta, uma tela para respostas estará disponível. Nesta janela o administrador é capaz de responder uma dúvida postada. Note novamente que esse procedimento deve ser feito apenas em casos excepcionais.
- **Users:** essa janela tem duas funcionalidades básicas: incluir novos usuários e acompanhar os usuários logados no sistema. São apresentados todos os usuários da competição e um formulário para cadastramento de novos. Clicando sobre um usuário pode-se alterar seus dados ou removê-lo do sistema. Também é possível incluir novos usuários através de um arquivo de importação.
- **Problems:** interface para cadastrar os problemas da prova. Para incluir um problema, deve-se especificar os dados do mesmo no formulário apropriado. Pode-se ainda remover ou alterar dados dos problemas já cadastrados.
- **Languages:** interface para fazer o cadastramento das linguagens da competição. Existem duas formas para tal cadastramento: importando de arquivo ou digitando os dados das linguagens. É possível ainda nesta janela alterar os dados da linguagens já cadastradas ou excluí-las.
- **Site:** neste item podem ser visualizados (e alterados) os dados relativos ao site. As principais informações são: nome, endereço IP, dias/horários da competição, regras de submissões e placares, quantidade de registros atualmente no sistema. Nesta janela pode-se iniciar/terminar uma competição, remover definitivamente submissões, dúvidas, tarefas do sistema e controlar globalmente o acesso dos usuários.
- **Contest:** nesta pasta podem ser vistos e/ou alterados os dados gerais da competição ativa. Os dados principais são nome da competição, dias e horários, regras de placar, recebimento de respostas de submissões e penalidades.

- **Logs:** nesta pasta são mostrados todos os registros de ocorrências. É possível ainda alterar a ordem e fazer consultas específicas nos registros existentes.
- **Tasks:** aqui é feita a administração das tarefas que estão sendo executadas pelo pessoal de staff. É possível gerenciar completamente as tarefas.
- **Answers:** nesta pasta o administrador deve configurar quais as respostas possíveis que um juiz dará para uma submissão. Apenas uma dessas respostas pré-definidas poderá ser enviada para o time como motivo para seu programa estar certo/errado.
- **Reports:** nesta pasta o administrador pode gerar relatórios da competição (como placar final, lista de equipes, lista de submissões, etc) e importar ou exportar uma competição.
- **Options:** nesta pasta é possível alterar os dados do usuário administrador.
- **Logout:** opção para desconectar do sistema.

Assim, para colocar uma competição em funcionamento, praticamente todo o serviço necessário está ligado com a interface do administrador, que deve incluir problemas, respostas, linguagens, usuários, e configurar os dados da competição e do site.

3.4. Staff

BOCA		Username: Pessoa do staff (site=2)		110 minute(s) left			
Tasks (4)		Score		Options		Logout	
Task #	Time	User / Site	Description	File	Status	Actions	
1	1	team1(1) / 2	Staff assistance		opentask	get	
3	1	team1(1) / 2	File to print	Autoexec.bat	opentask	get	
5	83	team1(1) / 2	Delivery to "team1" a balloon for problem Problema 3:		opentask	get	
6	98	team2(2) / 2	Delivery to "team2" a balloon for problem Problema 1:		opentask	get	

Figura 8: Tarefas que o pessoal de staff tem que executar.

As principais funções do pessoal de *staff* durante a prova são:

1. Controlar o acesso e movimentação dos times no local da prova.
2. Imprimir arquivos solicitados pelos times e entregar a impressão aos mesmos.
3. Entregar balões para cada time que acerta um problema (balões são usados como "placar visual" durante as competições).
4. Dar assistência a times com problemas de hardware e/ou software.

Apesar de todos estes itens necessitarem de alguma interação física, o controle sobre os pedidos de impressões, entregas de balões e pedidos de assistência podem ser feitos pelo sistema BOCA. Assim, o ambiente disponível para o pessoal de staff possui interfaces para alterar seus dados, verificar o placar durante a prova, e pegar tarefas para processar. As tarefas necessárias são basicamente aquelas descritas anteriormente (com exceção à primeira, onde o sistema não ajuda muito). Assim, uma pessoa do staff tem as opções de pegar tarefa para fazer, marcar tarefa como feita ou ainda devolver uma tarefa para que outra pessoa do staff a faça.

BOCA		Username: Placar (site=2)				117 minute(s) left	
Score				Logout			
#	Username	User Site	User	Solved	Problem Details	Total Time	
1	team2	2	Universidade B	1	Problema 1	98	
2	team1	2	Universidade A	0		0	

Figura 9: Placar online durante a competição.

3.5. Placar

O objetivo básico da existência de um usuário placar é poder disponibilizar pela rede o resultado em tempo real da competição para pessoas que não estão diretamente envolvidas com ela. Assim, este ambiente do placar não possui nenhum tipo de opção de configuração e/ou alterações no sistema. Ele é apenas para leitura dos resultados parciais da competição por pessoas locais que não devem ter muito acesso (como os técnicos das equipes), ou por pessoas localizadas fora do ambiente da prova.

3.6. Competições distribuídas e treinamento online

Devido ao uso de interfaces web, o sistema BOCA pode ser usado para a criação de competições distribuídas, onde equipes podem estar espalhadas em qualquer lugar com conexão à internet. Claro que competições como a Maratona de Programação ou o ICPC da ACM não poderiam ser feitas simplesmente dessa forma (pois todos teriam acesso à internet), mas simulações e outras competições poderiam ser enquadradas, como as que ocorrem no famoso site da Universidade de Valladolid [13], acessado diariamente por milhares de estudantes de computação de todo o mundo.

Site number: (local site)
 Name:
 Public IP:
 Start date (contest=20:51:52 BRT - 06/Apr/2004): hh:mm : dd/mm/yyyy / /
 End date (contest=01:51:52 BRT - 07/Apr/2004): hh:mm : dd/mm/yyyy / /
 Duration (contest=300): 300
 Stop answering (contest=270):
 Stop scoreboard (contest=240):
 Runs/clars that will be judged here (comma sep):
 Tasks that will be treated here (comma sep):
 Active:
 Autoend:
 Global score:
 Logins are: enabled
 Score level:
 Number of Clars: 2
 Number of Runs: 5
 Number of Tasks: 6

Figura 10: Configurações disponíveis em um site para uma competição distribuída com correção centralizada.

Além disso, o BOCA tem a opção de ser usado para uma competição com diversas sedes (ou sites), todos trabalhando ao mesmo tempo e com correção centralizada das submissões e das dúvidas. Isso torna viável uma competição em larga escala atingindo pontos extremos de um país como o Brasil, como já podemos notar na Maratona de

Programação. Apenas com processamento centralizado das submissões e dúvidas podemos ter igualdade de condições para todos os participantes, independente do lugar onde os mesmos estiverem, pois tudo será processado pelo mesmo grupo de juízes.

3.7. Segurança e Confiabilidade

Do ponto de vista de invasões, a maioria das transferências de dados são feitas através do protocolo de páginas HTTP, e assim podemos utilizar sua versão segura, o HTTPS. As demais são feitas entre bancos de dados PostgreSQL, que também aceita conexões seguras com autenticação e SSL. Além disso, o uso de um sistema operacional adequado e softwares de firewall (para bloqueio de conexões indesejadas) torna o ambiente do BOCA completamente seguro para uma competição distribuída.

Para garantir o funcionamento do sistema durante toda a competição, dependemos de diversos fatores externos e de hardware. A escolha de manter todos os dados dos times no servidor garante que, em caso de problema de hardware ou queda de força em um computador de um time, tudo que ele tem salvo está armazenado no servidor. Assim, mesmo que seja necessário alocar outro computador para este time, basta que esta nova máquina já esteja preparada para ser usada na competição, e o time poderá continuar sua prova com um tempo mínimo de prejuízo.

Dessa forma dois pontos de preocupação ainda restam: os servidores e a conexão entre servidores remotos (no caso de competições distribuídas). Para aumentar a confiabilidade dos servidores, podemos usar máquinas com recursos de hardware redundantes, e manter sempre um computador equivalente à disposição para troca rápida se o problema for mais grave. Já com relação à conexão, não há muito o que fazer. O sistema BOCA é projetado para utilizar a conexão sempre que a mesma estiver ativa, e desistir da mesma quando não é possível prosseguir. Assim, caso um servidor perca sua conexão com os demais, nenhum dos outros é afetado (a menos pelo fato de não receber mais informações atualizadas daquele servidor), e a competição pode continuar normalmente, agora com a necessidade de duas equipes de juízes (uma para cada parte desconexa da rede).

4. Conclusões

Neste trabalho apresentamos o Maratona Linux e o BOCA: sistema que vem sendo utilizado no gerenciamento de competições de programação. O sucesso de olimpíadas científicas reside em incentivar alunos dos cursos a estudar para obter bons resultados nas competições. O formato destes concursos de programação vem sendo utilizado também em disciplinas regulares, uma vez que permite aos alunos uma resposta imediata às soluções para os problemas por eles propostas, assim como acompanhar o desempenho de seus colegas. Esperamos que o sistema BOCA venha a ser utilizado em competições por todo o país, bem como em apoio a disciplinas com tais características. Estas competições vêm experimentando crescimento extraordinário e despertando cada vez mais o interesse nos alunos de cursos de computação do Brasil. Este interesse é bastante positivo, uma vez que motiva os alunos a estudar importantes conceitos da computação, como estruturas de dados, análise de algoritmos, técnicas e linguagens de programação, etc.

Referências

- [1] ACM International Collegiate Programming Contest, *The ACM Programming Contest Website*, <http://icpc.baylor.edu/icpc>
- [2] The Apache Software Foundation, *Apache HTTP Server Project*, <http://httpd.apache.org>

- [3] C. P. de Campos, *BOCA Home Page*, <http://boca.incubadora.fapesp.br>
- [4] E. Geschwinde, H. Schoenig, *PHP and PostgreSQL Advanced Web Programming*, Sams, 2002
- [5] International Olympiad in Informatics, *IOI Home Page*, <http://olympiads.win.tue.nl/oi>
- [6] Maratona de Programação da SBC, *Maratona Home Page*, <http://maratona.ime.usp.br>
- [7] Olimpíada Brasileira de Informática da SBC, *OBI Home Page*, <http://olimpiada.ic.unicamp.br>
- [8] The PHP Group, *PHP Hypertext Processor Home Page*, <http://www.php.net>
- [9] The PostgreSQL Global Development Group, *PostgreSQL Database System Home Page*, <http://www.postgresql.org>
- [10] D. Thomas, *Professional PHP4 Programming*, Wrox Press Inc, 2002
- [11] Red Hat, Inc, *Red Hat Home Page*, <http://www.redhat.com>
- [12] S. S. Skiena, M. Revilla, *Programming Challenges*, Springer Verlag, 2003
- [13] Universidad de Valladolid, *Valladolid Programming Contest Site*, <http://acm.uva.es>