



10º Congresso de Pesquisa

MIDDLEWARE PARA MANIPULAÇÃO E PROCESSAMENTO DE IMAGENS EM UM CLUSTER DE DISPOSITIVOS EMBARCADOS

Autor(es)

JOSE LUIS ZEM

Co-Autor(es)

BRUNO TEODOSIO GONÇALVES

1. Introdução

A possibilidade de transformar, computacionalmente, imagens em informações permitiu uma maior aproximação do homem com a máquina uma vez que o primeiro é orientado, principalmente, pela visão. Ao longo do tempo muitos algoritmos foram desenvolvidos a fim de viabilizar o entendimento de uma imagem pelo computador, porém tais algoritmos eram extremamente complexos e necessitavam de uma capacidade computacional considerável, muitas vezes só obtido com o uso de computadores especialmente projetados para este fim.

Atualmente os computadores pessoais aumentaram consideravelmente a capacidade de processamento, principalmente com a utilização de placas específicas para processamento de vídeo, porém esse aumento de poder computacional tem um preço e montar um cluster [6] [11] com computadores com estas características pode ter um custo considerável. Esse avanço tecnológico se reflete nos sistemas embarcados, permitindo que o processamento de imagem possa ser executado em dispositivos pequenos e de baixo custo.

Um cluster de computadores tem como principal característica a de se comportar, na visão do usuário, como se fosse um único sistema lógico, apesar de ser constituído, fisicamente, por vários computadores; contudo o aplicativo que irá trabalhar neste ambiente deve dividir o processamento de maneira que o mesmo possa ser balanceado ou distribuído entre os nós que compõem o referido ambiente [7].

A plataforma MINI2440, da Friendly ARM, não possui um grande poder computacional, porém tem como principais características o baixo custo, o tamanho reduzido e o baixo consumo de energia [3], quando comparado a um computador pessoal, justificando assim sua escolha como hardware básico para o cluster.

A manipulação de imagens (com grande volume de informação) continua, ainda, uma tarefa complexa e muitas vezes demorada, em razão da grande quantidade de operações necessárias para manipulá-las.

A biblioteca OpenCV é uma das mais utilizadas dentre as bibliotecas de manipulação de imagens. Isso em razão de estar em constante desenvolvimento pela comunidade que a mantém e também por ser gratuita, não tendo restrições com relação à direitos autorais e permitindo modificações em seus códigos [1].

Considerando o exposto, o trabalho de pesquisa em questão utilizou-se da biblioteca OpenCV [5], a qual permite a manipulação e processamento de imagens de forma simplificada, aplicada à plataforma MINI2440, que possui um poder computacional reduzido, porém de baixo custo e com tamanho reduzido, para a construção de um cluster de computadores com objetivo específico e utilizando-se de sistemas embarcados [4] [9] [10]. Também foi desenvolvido um middleware [2] [8] específico para processamento de imagens no referido cluster.

2. Objetivos

O trabalho de pesquisa desenvolvido e aqui relatado teve como objetivo principal o desenvolvimento de um middleware específico para processamento de imagens, utilizando-se a plataforma MINI2440, organizada em um cluster de computadores.

3. Desenvolvimento

O cluster desenvolvido era composto por três plataformas MINI2440, da Friendly ARM, e um computador pessoal que atuou como servidor de arquivos (necessário para que exista um diretório comum entre as três plataformas MINI2440 e também em razão de que tal sistema embarcado possui espaço de armazenamento bastante limitado), conforme pode ser observado na Figura 01. Todos os nós do cluster utilizaram o sistema operacional baseado no kernel Linux, porém o computador pessoal usou uma versão convencional do referido sistema operacional (distribuição Ubuntu) e os demais nós empregaram uma versão para preparada para sistemas embarcados.

As MINI2440 e o computador pessoal foram interligados utilizando um switch FastEthernet e cabos par trançado. O computador pessoal compartilhou um diretório, através do sistema de arquivos NFS (Network File System), sendo que tal diretório foi utilizado como ponto comum entre os nós, ou seja, permitindo que todos tivessem acesso aos arquivos gerados pela aplicação de processamento de imagem.

Cada MINI2440 possui 100 x 100 mm de diâmetro e, assim, ocupam pouco espaço; já o computador pessoal utilizado nessa pesquisa foi um computador de mesa (desktop). Em casos onde espaço é fundamental o computador de mesa pode muito bem ser substituído por um notebook, desde que as configurações permaneçam as mesmas.

O desenvolvimento do middleware para o processamento das imagens foi dividido em quatro partes (o protocolo de identificação entre os nós, o protocolo de comunicação entre nós e a biblioteca utilizada por aplicações que utilizarão o cluster), o que resultou em quatro softwares. A interação entre tais softwares pode ser verificada na Figura 02a.

Na primeira parte do desenvolvimento do middleware, foi elaborado um software para realizar a identificação entre os nós. Esse software realiza a troca de mensagem entre os nós para identificar a quantidade e o endereço dos mesmos. Para que um nó faça parte do sistema, é necessário que o mesmo execute uma instância do identificador, visto que, essa instância é responsável por identificar todos os nós desse sistema e informar aos mesmos que existe um novo nó. O identificador realiza a contagem dos nós na rede, sendo que, para cada resposta recebida, incrementa em um o contador de nós.

Na segunda parte do desenvolvimento do middleware, foi elaborado um software para enviar e receber mensagens da aplicação (chamada aqui de aplicação usuária) que utilizaria o cluster. Esse software, chamado de comunicador, é responsável por registrar a aplicação usuária, receber e enviar mensagens da aplicação usuária e receber e enviar mensagens de outro comunicador. O comunicador deve ser executado após o identificador sendo que, ao iniciá-lo, ficará esperando por mensagens provenientes de outro comunicador ou requisições de envio vindas da aplicação usuária.

Para que uma aplicação usuária possa registra-se no sistema e enviar e receber mensagens, foi desenvolvida uma biblioteca, a qual permite que uma aplicação usuária tenha acesso ao comunicador e possa utilizar o cluster. Essa biblioteca oferece funções para se registrar uma aplicação usuária, enviar mensagens para um nó, receber mensagens de um nó e requisitar o número de nós presentes no cluster.

Como último passo do desenvolvimento do middleware, foi implementada uma aplicação que se utilizou da biblioteca para processar imagens no cluster. Essa aplicação teve como objetivo único testar e demonstrar o uso do middleware para processamento de imagem no cluster.

A aplicação usuária deve ser executada em todos os nós, porém, somente um deles é considerado como mestre, as demais instâncias são consideradas como escravas, cabendo ao mestre distribuir tarefas entre os nós escravos.

Quanto à comunicação entre processos, a troca de mensagem entre o identificador e o comunicador é realizada através de fila de mensagens, ou seja, é criada uma fila de mensagens na memória onde as mensagens são escritas e lidas pelas aplicações.

Ao inicializar o identificador, ele cria uma fila de mensagem, pela qual receberá mensagens do comunicador. Ao inicializar o comunicador, este cria uma fila de mensagem para receber comandos da biblioteca e outra para receber mensagem do identificador e utiliza a fila criada pelo identificador para enviar comandos para o mesmo.

Ao iniciar uma aplicação, utilizando a biblioteca, cria-se uma fila de mensagem pela qual irá receber mensagens do comunicador e utiliza a fila criada pelo comunicador para enviar comandos a ele.

A interação do middleware, bem como os seus módulos executando em cada um dos nós, pode ser observada na Figura 02b.

4. Resultado e Discussão

Foram realizados dois testes, sendo que, um deles trabalhou com o processamento de uma imagem, representada pela Figura 03a, utilizando-se o cluster e o outro teste efetuou o processamento da imagem em um único nó.

No caso do primeiro teste, cada nó do cluster recebeu uma cópia da imagem e aplicou em uma área específica da mesma um procedimento matemático de alteração dos pixels, gerando uma nova imagem com as alterações, porém apenas na parte que lhe cabia.

A junção das partes e criação a imagem final (Figura 03b) é de responsabilidade do nó eleito como mestre (uma das plataformas MINI2440). No caso do segundo teste, um único nó manipulou localmente toda a imagem, gerando a nova imagem (Figura 03c) contendo as alterações.

Como resultado, percebeu-se que, as Figuras 03a e 03b deixaram isso perceptível, cada nó realmente trabalhou, única e exclusivamente, na parte que lhe cabia da imagem. Também através da execução pode-se verificar que todos os softwares desenvolvidos (o identificador, o comunicador, a biblioteca e a aplicação usuária) comportaram-se conforme o planejado e programado, enviando e recebendo corretamente as mensagens de controle, sincronização e de comunicação, bem como com os resultados dos processamentos.

Em suma, tanto a parte relativa à preparação do hardware do cluster como o software funcionaram corretamente, executando aquilo que se esperava deles, concluindo-se assim, que todo o desenvolvimento cumpriu com os objetivos definidos.

5. Considerações Finais

Quanto às considerações finais figura-se a proposição de refazer o experimento aqui relatado, porém com a perspectiva de comparar o tempo gasto para o processamento distribuído da imagem através do middleware em um ambiente composto por computadores de uso geral (domésticos), além de recriar a estrutura utilizando um número maior de MINI2440.

No primeiro caso para se determinar o quão próximo ou distante está o desempenho da MINI2440 quando comparado ao de um computador de uso geral e no segundo caso para verificar se a comunicação entre tais sistemas embarcados afeta o desempenho da aplicação.

Referências Bibliográficas

[1] BRADSKI, Gary; KAEHLER, Adrian; Learning OpenCV: Computer Vision with the OpenCV Library. 1ª Edição. Editora O'REILLY. 2008.

[2] COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim; Sistemas Distribuídos: Conceitos e Projetos. 4ª Edição. Porto Alegre: Editora Bookman, 2007.

[3] Documentação retirada do site do fabricante. Disponível em: <http://www.friendlyarm.net/products/mini2440>. Acessado em: 22 Fev. 2012

[4] HALLINAN, Christopher; Embedded Linux Prime: A Pratical Real-World Approach. 1ª Edição. Editora Prentice Hall, 2007.

[5] LAGANIÈRE, Roberto; OpenCV 2 Computer Vision Application Programming Cookbook; 1ª Edição. Editora Packt Publishing, 2011.

[6] PITANGA, M.; Computação em Cluster: O estado da Arte. 1ª Edição. Rio de Janeiro: Brasport, 2003

[7] TANENBAUM, Andrew S.; Distributed Operating Systems. 1ª Edição. New Jersey: Editora Prentice Hall, 1995.

[8] TANENBAUM, Andrew S.; STEEN, Maarten Van. Distributed Systems: Principles And Paradigms. 1ª Edição. New Jersey: Editora Prentice Hall, 2002.

[9] VAHID, Frank; GIVARGIS, Tony; Embedded System Design: A Unified Hardware/Software Approach. In: Department of Computer Science and Engineering University of California. Disponível em:

Anexos

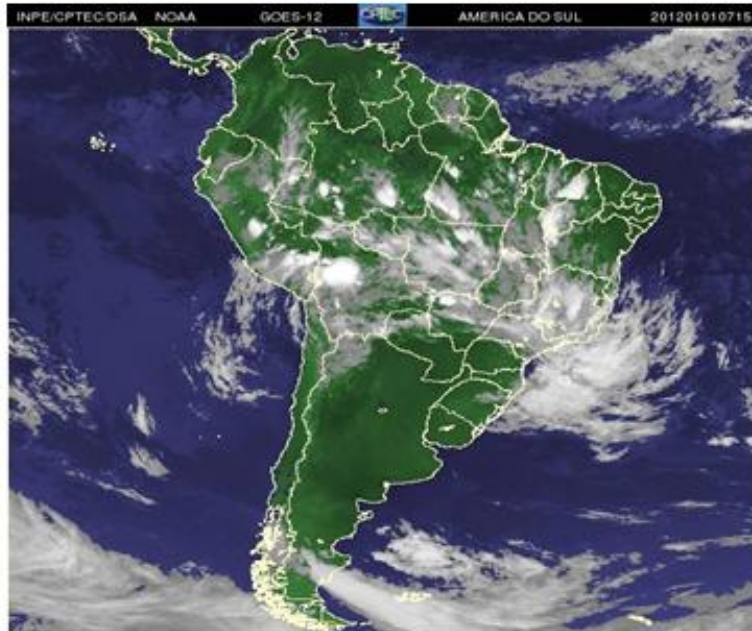


Figura 03a: Imagem original.

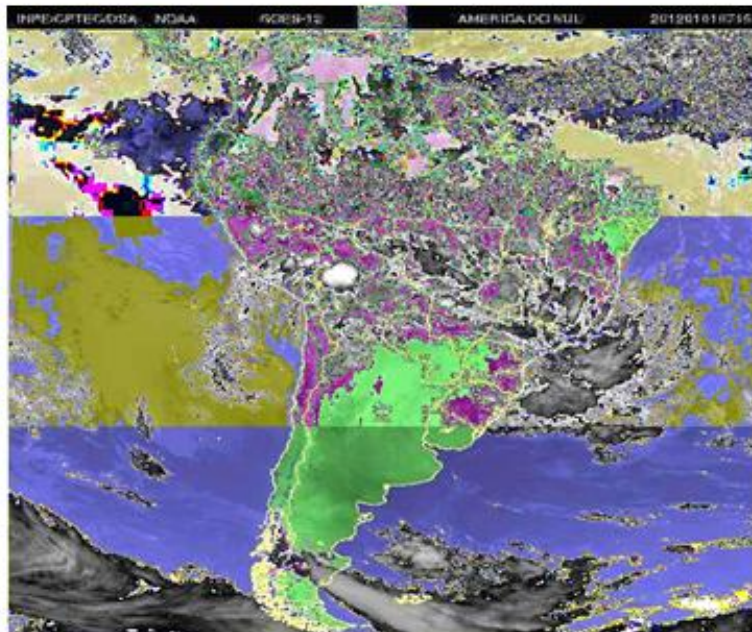
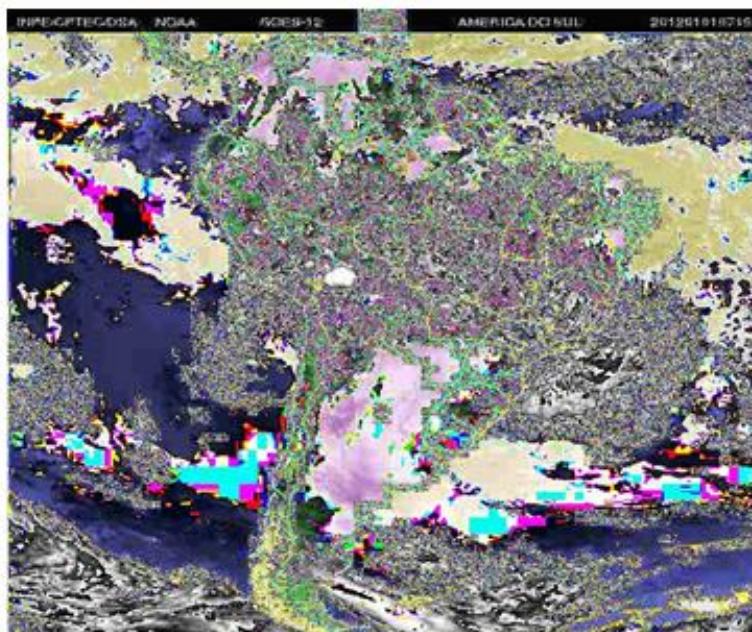


Figura 03b: Imagem processada por três nós.



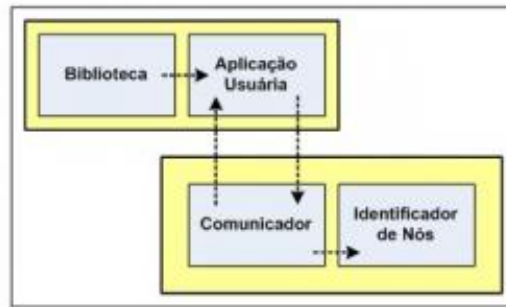


Figura 02a: Interação entre os módulos de software do middleware.

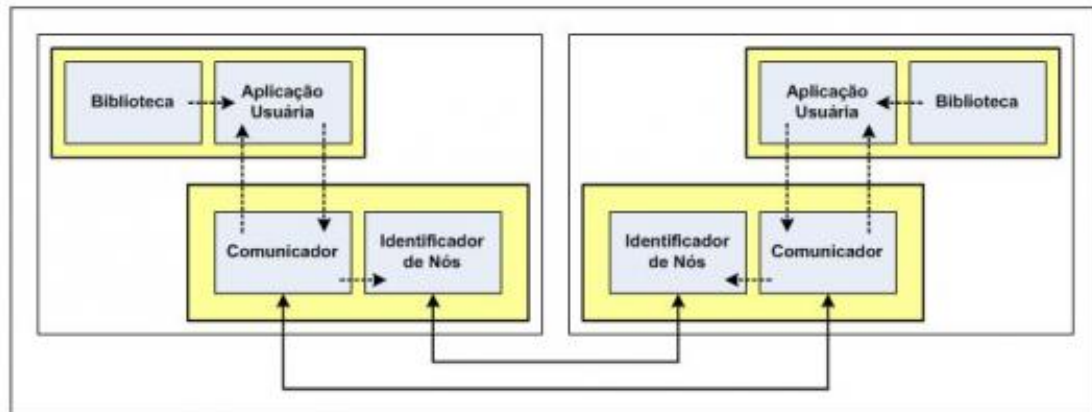


Figura 02b: Interação entre os módulos de software do middleware executando em nós diferentes.

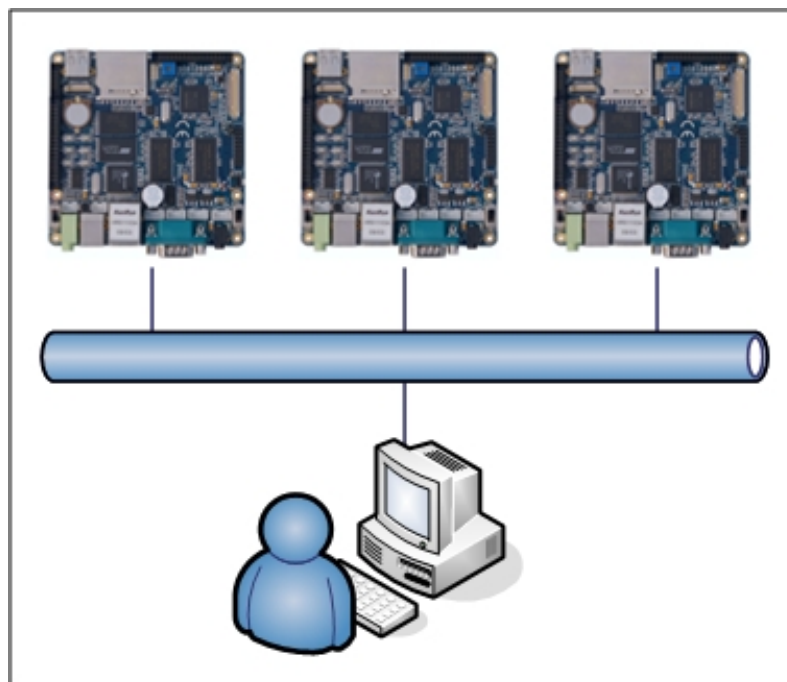


Figura 01: Cluster de computadores usado nos experimentos.