

O Desafio da Universalização de Acesso à Internet: O Projeto do Computador Popular

Dorgival Guedes^{1*}, Fabrício Konzen¹, Guilherme Ribeiro¹,
Lamarque Souza¹, Sérgio Campos¹, Wagner Meira Jr.¹

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte MG Brasil 31270-010

luar@dcc.ufmg.br

Abstract. *Universal access to the Internet has been proposed as an efficient strategy to minimize the social divide and to grant access to technological advances and information to underprivileged classes. However, a major obstacle in this process is the cost of the computational platform, both in terms of software and hardware. In this article we present the popular computer project developed at the DCC-UFMG. The main aspects of the design are presented, the major technical challenges faced are discussed and the architecture (both software and hardware) is described with an emphasis on the development of an efficient, integrated system.*

Resumo. *A universalização do acesso à Internet tem sido apontada como uma forma eficaz de diminuir as desigualdades sociais e promover camadas da população que sempre foram discriminadas nos avanços tecnológicos. Entretanto, uma grande barreira a esse processo é o custo da plataforma, tanto em termos de hardware quanto software. Neste artigo apresentamos o projeto do computador popular desenvolvido pelo DCC-UFMG. Os principais detalhes do projeto são apresentados, os principais problemas técnicos enfrentados são discutidos e a arquitetura de software e hardware é descrita com ênfase na criação de um sistema integrado e eficiente.*

1. Introdução

Em seu nível mais básico, acesso universal significa garantir a todos os cidadãos acesso aos serviços de telecomunicações; em particular, acesso universal à Internet implica no acesso amplo, irrestrito e de baixo custo à rede mundial de computadores. O grande desafio é que as disparidades e desigualdades hoje existentes — usualmente associados ao termo "exclusão digital"(ou *Digital Divide*, no inglês) — são consideráveis, particularmente em países em desenvolvimento, onde os custos associados ao acesso à Internet a tornam inacessível para uma grande parte da população e mesmo para muitas instituições públicas. Tornando o problema ainda mais difícil, tais serviços muitas vezes estão disponíveis apenas em grandes centros.

*Financiado pela FUNDEP e CNPq

O Brasil, que desponta como uma das grandes nomes no campo da Internet e tecnologias afins entre os países em desenvolvimento, tem na universalização do acesso à Internet um grande desafio a ser vencido. Para a superação deste desafio são necessárias ações em várias frentes, como tarifas telefônicas diferenciadas, provimento amplo de acesso, desenvolvimento de uma cultura digital e uma plataforma computacional de custo acessível para a população. Em suma, o acesso universal pressupõe três elementos básicos: acesso físico à rede, existência de pontos de acesso (computadores) em larga escala e a disponibilização de informação de interesse do cidadão.

A veiculação de informação na rede é um fator importante que vem recebendo atenção do governo, que prepara programas nesta área, e da sociedade em geral. Com certeza, esse processo será fortemente acelerado com o aumento do número de pessoas com acesso à rede.

Para viabilizar a disponibilização de pontos de acesso em larga escala é imprescindível que as soluções sejam de baixo custo e de fácil operação, uma vez que o objetivo é permitir o acesso de cidadãos de todos os seguimentos da sociedade. Apesar disso, os dispositivos devem ter capacidade computacional compatível com a exigida para visualização dos vários tipos de conteúdo hoje disponíveis na rede, a fim de garantir o acesso pleno àquela.

Para se garantir acesso físico é preciso que haja uma infra-estrutura de rede de comunicações que permita o acesso do cidadão em todo o país, seja através de linhas discadas residenciais ou linhas de acesso colocadas em instituições públicas como escolas e bibliotecas.

Relacionada à questão da infra-estrutura está a implantação de servidores e outros dispositivos de rede que garantam as condições de operação dos equipamentos de acesso, uma vez que uma das melhores formas de se conseguir tais equipamentos a um custo baixo é transferindo parte das tarefas de processamento usuais para servidores especializados.

Para solucionar estas questões é preciso desenvolver soluções criativas para o cenário nacional. O Laboratório de Universalização de Acesso à Internet (LUAR) do Departamento de Ciência da Computação da UFMG tem por objetivo estudar e propor soluções em todas as áreas que se relacionam a este problema, algumas das quais serão discutidas neste artigo. O LUAR vem trabalhando nas áreas de desenvolvimento de soluções computacionais de baixo custo com interfaces simples de operação e dos servidores necessários à sua implantação, bem como avaliando as questões de redes de interconexão (infra-estrutura) diretamente relacionadas ao problema.

O restante deste artigo está dividido da seguinte forma: introdução. A próxima seção discute algumas iniciativas semelhantes em outros países. A seção 3. descreve a nossa solução, seguida de uma discussão sobre os desafios enfrentados na seção 4.. A seção 5. apresenta os projetos em execução, seguida de algumas conclusões e referências.

2. Trabalhos relacionados

Quando se fala em universalização de acesso, tem-se em mente a garantia de acesso a todas as camadas sociais. Se pensarmos que grande parte da população não está familiarizada com a informática e não possui meios ou não quer adquirir um computador pessoal

para tanto, concluiremos que existe a necessidade de uma solução de baixo custo e tão fácil de se utilizar quanto um eletrodoméstico.

Algumas soluções têm sido criadas nesse sentido [8, 14, 3, 19]. Todas possuem teclado, monitor e modem. Soluções não tão populares (em parte devido ao custo mais elevado) utilizam monitor LCD, *touchpad* ou *touchscreen*, como o Dot.Station [8]; alguns funcionam com teclados sem fio, como a WebTV [14]; outras suportam monitores SVGA comuns. Geralmente outras opções de conexão como DSL e até via cabo de TV são oferecidas. Com relação ao processador, a maioria se utiliza de algum processador compatível com o Intel Pentium (AMD K6, Cyrix MII, Geode, etc). A quantidade de memória RAM se situa entre 32 e 64MB. O sistema operacional é armazenado em memória *flash*, EEPROM ou mesmo CDROM [2]. Enquanto alguns utilizam sistemas operacionais embutidos especiais [7], outros utilizam Windows CE [11], e há soluções que adotam Linux como sistema operacional.

Existem soluções que permitem a utilização de qualquer provedor e outros que obrigam a utilização de provedores específicos associados ao fabricante do computador. Os preços (com monitor) variam entre US\$ 300 e US\$ 600 nos Estados Unidos; em alguns casos há descontos/reembolso na adoção de um provedor específico por um determinado período de tempo.

Apesar dessa variedade, os modelos encontrados não atendem aos objetivos deste projeto por problemas variados: aqueles que utilizam provedores específicos normalmente utilizam também *software* proprietário, não permitindo acesso irrestrito; outros não atingem o preço desejado. Nenhuma solução aceitável existe, em particular, no mercado nacional. O protótipo do Computador Popular do LUAR tem por objetivo demonstrar que há soluções viáveis que podem ser levadas a cabo no país.

3. A Solução LUAR

O protótipo desenvolvido possui dois modos de funcionamento: conectado a um servidor em uma rede local ou operando de forma independente. Sem estar conectado a um servidor o usuário possui somente as funcionalidades de acesso à Internet e edição de textos simples. Não é possível salvar arquivos na memória *flash* enquanto o sistema está executando, pois ela é montada como somente leitura.

Como solução para armazenar arquivos de usuários temos o acoplamento de dispositivos de armazenamento, tais como disco rígido, disquete, ou mesmo o uso de uma partição da memória *flash* ou uma segunda unidade. Pode-se também armazenar os dados na própria Internet através de provedores (*ASP, Application Service Providers*) ou em um servidor criado especificamente para esse fim. Em nosso trabalho consideramos todas estas opções a fim de poder oferecer diferentes soluções aos fabricantes de *hardware*, que podem então optar por aquela que melhor se adequa à sua estratégia de mercado. Consideramos, entretanto, que as melhores soluções são o uso da memória *flash* e servidores especiais, que podem oferecer uma combinação bastante flexível.

O uso eficiente da memória *flash* cria um espaço de armazenamento local na máquina que pode ser utilizado mesmo quando se sem conexão a qualquer rede. O acesso a um servidor especial através de um protocolo de acesso a arquivos remotos oferece

uma solução para áreas de armazenamento maiores e que permite ao usuário acessar seus dados independentemente do computador utilizado.

Através de um servidor é possível também acessar outros programas, tais como aplicativos de escritório (editores/formatadores de texto, planilhas, etc.), aplicativos multimídia (exibidores de vídeos e processadores de áudio) e qualquer outro programa que esteja disponível no servidor.

O protótipo é capaz de identificar automaticamente se existe um servidor apropriado instalado em uma rede local, quando então passa a compartilhar o diretório **/usr/local** com o servidor através do protocolo NFS (*Network File System*). Este protocolo se adapta bem para aplicações em redes locais. Quando o computador é utilizado isoladamente, conectando-se através da linha telefônica, por outro lado, esta não é uma solução viável para o acesso a dados no servidor remoto que se encontra do outro lado da linha. A sessão 5. apresenta uma proposta de implementação de um protocolo mais adequado a esta tarefa.

3.1. Casos de uso

O protótipo, como apresentado, oferece várias possibilidades de aplicação, contemplando cenários bastante diversos.

3.1.1. Uso residencial

Este foi o primeiro objetivo do projeto quando de sua criação, uma vez que um computador de baixo custo pode ser acessível a uma grande parcela da população até então excluída por fatores financeiros. Os usuários serão beneficiados através do provimento de informações de lazer, através do exercício da cidadania por meio da utilização de serviços providos pelo governo e outros órgãos da sociedade e através da educação à distância que pode ser provida por instituições de ensino.

Em seu uso residencial o computador popular opera como um terminal de acesso à Internet, oferecendo recursos de um *browser* completo. Na sua versão atual o equipamento permite que o usuário trabalhe desconectado da rede para geração de documentos locais utilizando os programas instalados na *flash*, o que oferece maior flexibilidade de utilização. A partir da conexão à Internet o usuário pode então ter acesso a todos os serviços da rede e de uma área de armazenamento remota de maior capacidade. Neste caso o acesso a aplicações providas remotamente se torna menos provável, dado o custo de transmissão dos arquivos de programa de grandes aplicações através da rede telefônica.

3.1.2. Escolas e empresas

Com o desenvolvimento do projeto viemos a considerar que os cenários mais interessantes para a utilização do computador popular podem ser em ambientes escolares ou corporativos, onde o computador tem aplicação direta no apoio à educação ou como instrumento de trabalho. Neste caso, com a disponibilização de um grande número de computadores populares em um mesmo local torna-se razoável a adição de um computador mais completo que opere como servidor, expandindo enormemente a capacidade dos computadores populares por um custo adicional por unidade relativamente baixo.

Consideremos uma escola pública com aproximadamente 400 alunos ou uma pequena empresa. Tais ambientes poderiam se valer de um conjunto de vinte máquinas cliente (computadores populares) e um servidor interligados por meio de uma rede local. Chamamos a atenção aqui para o fato de que o servidor não precisa ter uma configuração muito diferente dos clientes, apenas recebendo a adição de periféricos comuns: disco rígido, CD-ROM e disquete.

Um ponto importante nesta configuração é a redução drástica de custo de implantação e manutenção do ambiente. Onde gastava-se R\$ 1.500,00 por máquina potencialmente podemos chegar a R\$ 700,00 por cliente. A colocação dos dados dos usuários no servidor favorece uma política de *backup* centralizada. A máquina cliente, por não usar disco rígido ou qualquer outra parte móvel, pode ter a garantia e a durabilidade de uma televisão. Adotando-se o LINUX como solução de sistema operacional, economiza-se também no *software*, pois além das vantagens econômicas, o LINUX apresenta uma gama significativa de aplicativos gratuitos que poderão atender às mais diversas necessidades de aplicações educacionais ou empresariais.

3.1.3. Quiosques

Uma tendência em vários centros urbanos é disponibilização de pontos de acesso à Internet instalados em locais estratégicos da cidade: farmácias, *shopping centers*, centros comunitários, igrejas, etc.. Assim como os telefones públicos, os quiosques, como são conhecidos, poderão atender a uma parcela da população que não tem interesse em comprar um computador ou não pode fazê-lo. Nesta configuração, o custo é um fator importante. Uma empresa administradora desta solução pode duplicar sua base instalada caso adote uma solução semelhante ao computador popular. Neste caso os computadores podem ser configurados para se valer da rede proprietária de conexão dos quiosques para acessar servidores especialmente configurados.

3.2. Hardware

No início do projeto aventou-se a possibilidade de se basear a arquitetura em computadores antigos, o que permitiria o reaproveitamento de equipamentos abandonados e poderia reduzir o preço final. Apesar desta solução apresentar alguns aspectos interessantes e de certo apelo social, decidimos não seguir nesta direção por dois motivos principais: primeiramente, os usuários que viessem a se utilizar desses computadores teriam não só uma resistência psicológica ao equipamento, por se tratar de um *hardware* ultrapassado, mas também experimentariam problemas sérios ao acessar informação em páginas na Web, uma vez que a grande maioria dos sites de maior interesse popular hoje fazem uso de elementos visuais, animações e áudio que se encontram além da capacidade de computadores de gerações anteriores. Em segundo lugar, essa solução traria problemas logísticos enormes, pois os implementadores teriam que lidar com problemas como a identificação de partes defeituosas nos computadores reaproveitados, bem como a adequação do *software* a uma enorme variedade de plataformas, processadores, configurações de memória, periféricos, etc.

Por todos estes motivos optamos por utilizar uma plataforma com componentes atuais, se bem que obviamente não considerados “topo-de-linha”. O uso de componen-

tes comuns, de “penúltima geração”, por assim dizer, nos permite atacar o problema de custo aproveitando-nos do fenômeno de economia de escala, já que tais componentes são produzidos e comercializados em larga escala (a menos da memória *flash*, usualmente).

O protótipo montado inicialmente possui a seguinte especificação de *hardware*:

- Processador AMD K6-II de 500 MHz
- Placa mãe SiS 530 (arquitetura da placa de domínio público) com uma porta para impressora, conexão para até quatro dispositivos IDE (disco rígido, cdrom, etc), duas portas seriais e duas portas USB, teclado e mouse.
- 64 megabytes de memória RAM
- Flash disk de 16 Megabytes, tecnologia Disk-on-chip (785 kbyte/s de taxa de leitura média)
- Placa de vídeo Silicon Integrated Systems [SiS] 6306 3D-AGP (*on board*)
- Placa de rede Davicom Semiconductor, Inc. Ethernet 100/10 MBit (*on board*)
- Modem de 56 Kbps C-Media Electronics Inc CM8738 (*on board*)
- Placa de som C-Media Electronics Inc CM8738 (*on board*)
- Monitor 14 polegadas, colorido, não entrelaçado

Todo o hardware do protótipo exceto o *disk-on-chip* é comum a computadores pessoais atuais e pode ser adquirido em lojas convencionais de computadores. O *disk-on-chip* é a única parte que não é comum em computadores pessoais atuais, no entanto não é difícil adquiri-lo em lojas especializadas em equipamentos para sistemas embutidos. Outras tecnologias de memória *flash* também são possíveis, como por exemplo dispositivos *compact flash* que começam a se tornar mais comuns como expansões de memória para câmeras digitais e computadores de bolso (PDAs).

Na escolha do hardware levou-se em conta principalmente preço, funcionalidade oferecida e facilidade de configuração no sistema operacional Linux.

Outro objetivo do projeto foi a garantia da robustez do sistema. Partes móveis sofrem com desgastes inerentes ao atrito e por isso são mais suscetíveis a falhas. Tendo isto em mente decidiu-se eliminar todas as partes móveis do equipamento, ou seja, disco rígido, unidades de disquete e CD-ROM. Outras partes móveis, como as ventoinhas do processador e da fonte de alimentação não foram eliminadas no protótipo. No entanto, há várias soluções de fabricação possíveis utilizando-se um projeto especialmente desenvolvido para esse fim que podem eliminar tais elementos em uma versão industrial. Também objetivando a robustez do sistema, o *disk-on-chip* foi montado como somente leitura no primeiro protótipo. Desse modo um vírus de computador, por exemplo, não poderia alterar nenhum arquivo do sistema.

O disco rígido foi eliminado também por outro motivo. A capacidade dos discos rígido atuais excede em muito a necessidade de espaço de um usuário comum. Nesse sentido o uso de um dispositivo de menor capacidade, como a memória *flash*, além de diminuir o custo do sistema, também racionaliza o uso do espaço de armazenamento.

Devido ao espaço limitado e a ausência de uma área de acesso para escrita, não há recursos no equipamento para implementação de uma área de *swap* para memória virtual. O que se verificou na prática é que as versões atuais do sistema operacional (Linux) e dos aplicativos utilizados trabalham bem em 64 MB de memória RAM, sem necessitar de espaço extra para *swap*. O protótipo vem sendo utilizado em vários cenários sem que

a falta da memória virtual seja um problema, o que chegou a nos surpreender em certas situações.

3.3. Software

O foco principal do projeto do sistema foi dado ao *software*, pois tínhamos uma grande restrição de espaço de armazenamento determinada pelo tamanho da memória *flash* então economicamente viável, que era de 16 megabytes, aproximadamente. Por outro lado, não haviam grandes restrições ao *hardware* utilizado, que poderia ser qualquer um que fosse suportado pelo sistema operacional adotado (isto é, bastava que se o *hardware* tivesse os *device drivers* apropriados. Seguindo o objetivo de baixo custo do sistema e para garantir que poderíamos facilmente adaptar partes do sistema em caso de necessidade optamos por utilizar somente *software* livre (*free software*) e de código aberto. A importância do uso de *software* de código aberto nos deixava em condições de alterar os programas para as nossas necessidades.

Dentre os elementos do *software* armazenado no *disk-on-chip* do primeiro protótipo podemos destacar:

- Linux kernel 2.2.18
- XFree86 3.3.6
- ambiente gráfico KDE 2.0.1 (bibliotecas e *daemons*)
- Konqueror 1.9.8 (navegador Web e gerenciador de arquivos)
- KPPP (discador)
- KEdit (editor de textos simples)

O sistema operacional Linux [15] foi escolhido por ser de código aberto, possuir vasta documentação na Web e por oferecer a maior facilidade na obtenção de *device drivers* se comparado a outros sistemas operacionais de código aberto como o FreeBSD, por exemplo. Além disso, o Linux é mais conhecido no país e já possui uma base de usuários grande, facilitando assim a obtenção de suporte técnico especializado.

O principal aplicativo do sistema é o navegador Web. Existem vários navegadores de código aberto para Linux: Mozilla [16], Konqueror [20], Konqueror para sistemas embutidos [21], Nautilus [5] e Galeon [6], por exemplo. Outros navegadores como o Netscape [13] e Opera [18], por não serem de código aberto, não foram considerados inicialmente, apesar de serem gratuitos.

Nem todos implementam todas as funcionalidades que precisávamos de um navegador, como foi o caso de Galeon, Konqueror para sistemas embutidos, Nautilus e Opera que, por exemplo, não suportam *plugins* para áudio e vídeo no formato Real e Shockwave Flash, *applets* Java e nem SSL (*Secure Socket Layer*), utilizados em diversos *sites* Web. Dentre todas as opções, o Konqueror nos pareceu a melhor escolha por implementar todas as funcionalidades desejadas de um navegador Web (Java, Javascript, *plugins* para Shockwave Flash, Real, Cascade Style Sheet, SSL), além de ser um navegador rápido e mais “leve” que muitos dos outros que implementam as mesmas funcionalidades. Além disso, o Konqueror é também um gerenciador de arquivos). Outro ponto a favor do Konqueror é que ele faz parte de um ambiente gráfico totalmente integrado, o KDE [9]. Dessa forma poderíamos utilizar outros programas do mesmo ambiente economizando memória para as bibliotecas de interface gráfica, que seriam comuns a todos os programas. Existem

outros ambientes gráficos/gerenciadores de janelas para Linux, dentre eles o GNOME, WindowMaker, Enlightenment, mas nenhum deles possuía um navegador Web com as mesmas características do Konqueror integrado a eles.

No servidor instalamos o KOffice e o StarOffice como exemplos de aplicativos de escritório. O KOffice [10] possui a vantagem de ser integrado ao KDE e por isso não exigir memória extra para a interface gráfica (é importante lembrar que o sistema não possui memória virtual). Infelizmente, apesar do bom trabalho que vem sendo feito pela equipe de desenvolvimento do KDE, o KOffice ainda não pode ser considerado pronto para utilização em larga escala. Esperamos que até o fim do projeto o KOffice já esteja realmente estável e funcional para ser utilizado por qualquer usuário.

O StarOffice, por outro lado, é considerado o mais completo e estável aplicativo de escritório para Linux [12]. O principal problema desta solução é que o pacote traz todas as funcionalidades agrupadas, tornando-se um ambiente muito grande e computacionalmente pesado. A versão aberta do sistema, o OpenOffice [17], está em desenvolvimento e pretendemos incorporá-la ao sistema para resolver o problema.

4. Desafios do desenvolvimento do protótipo

Uma vez decidido que o sistema seria baseado em *hardware* tradicional ainda não obsoleto, utilizando memória *flash* para armazenamento e *software* aberto para o sistema operacional e os aplicativos, vários desafios ainda precisaram ser vencidos para tornar o protótipo realidade. Entre as mais importantes podemos destacar o desenvolvimento de uma interface de uso simples e intuitivo e a solução do problema de espaço.

4.1. Simplificação de uso

Escolheu-se o ambiente gráfico KDE por ele ser estável e possuir vários programas já integrados, inclusive o Konqueror. Uma das características dos ambiente para Linux é o alto grau de configurabilidade que possuem. Tal configurabilidade exige que o usuário faça escolhas para adequar o ambiente à suas necessidades. Como o público alvo deste projeto são pessoas que tiveram pouco ou nenhum contato anterior com computadores, foi necessário simplificar ao máximo a interface do ambiente. O objetivo era ter um sistema realmente simples que pudesse ser usado por qualquer pessoa e que dependesse o mínimo possível da interferência do mesmo pra poder funcionar.

Dentre as modificações podemos destacar a remoção de opções de configuração do sistema, a eliminação de programas que não fossem essenciais ao funcionamento do ambiente, eliminação do menu K (equivalente ao menu *iniciar* do Windows). A interface final possui um painel (chamado Kicker) com alguns botões para executar os programas instalados na memória *flash* e um relógio/calendário na sua extremidade direita. O navegador Konqueror é disparado quando o sistema é inicializado e está sempre em execução, se tornando o centro do ambiente do usuário. Se o usuário tenta terminar o Konqueror o sistema inteiro é reinicializado. Isto foi feito de forma que caso ocorra algum problema com o Konqueror e ele "morra", o usuário não precisará chamá-lo de novo. Mensagens de erro e confirmações associadas às mesmas foram removidas e o comportamento mais efetivo em cada caso é acionado automaticamente. Isto se justifica uma vez que esta não

é uma máquina para um usuário avançado que pode ter motivos para tomar decisões variadas em cada caso, mas sim para usuários leigos que apenas se confundiriam com o leque de opções apresentadas pelo sistema.

Outra modificação importante foi o uso da memória *flash* em modo somente leitura. Dessa forma prevenimos que vírus para Linux que por ventura possam surgir tenham a oportunidade de alterar o sistema. Outra vantagem dessa estratégia é simplificar o processo de ligar e desligar o aparelho. Como a área do sistema é acessível somente para leitura o usuário não precisa iniciar o processo de desligamento com a notificação ao sistema sobre o fato, como os sistemas tradicionais. Ao invés disto, o usuário pode simplesmente desligar o aparelho sem que nenhuma informação seja perdida. Essa estratégia também evita o uso do programa *fsck* (*filesystem checker*) durante o processo de *boot* caso o sistema tenha sido reinicializado por algum motivo, já que o sistema de arquivos não pode ser alterado.

4.2. Adaptação ao espaço disponível

Para podermos acomodar todos os programas necessários ao funcionamento do equipamento de forma satisfatória para os usuários tivemos que enfrentar um sério problema de espaço. Para resolvê-lo várias técnicas foram empregadas, tais como:

- eliminação de programas não essenciais ao sistema,
- enxugamento do núcleo do sistema operacional,
- compressão de sistema de arquivos,
- eliminação de símbolos dentro de arquivos objeto,
- diminuição do número de *inodes* do sistema de arquivos,
- substituição de *soft links* por *hard links*,
- eliminação de comentários e texto em outras línguas.

Eliminação de programas não essenciais, eliminação de comentários dentro de arquivos texto, eliminação de símbolos dentro de arquivos objeto, e enxugamento do kernel são as técnicas mais comuns para diminuir o tamanho de um sistema Linux. As duas primeiras são simples de serem entendidas. A eliminação de símbolos dentro de arquivos objeto foi obtida utilizando-se o comando *strip*. O Enxugamento consistiu em eliminar todas as funcionalidades e *device drivers* que não seriam utilizados.

Há dois tipos de *link* no Linux (estruturas que fazem a associação de nomes e arquivos), chamados de *hard* e *soft link*. Um *hard link* é apenas um nome que é associado diretamente ao arquivo, que pode ter vários nomes no Linux. Um arquivo só é apagado do sistema quando seu último nome for removido. Não existe conceito de nome original, todos os nomes possuem o mesmo valor para o arquivo. Geralmente, mas não necessariamente, todos os nomes de um arquivo são encontrados no sistema de arquivos que contém os dados do mesmo.

Um *soft link* (ou *link* simbólico) é uma entidade completamente diferente: ele é um arquivo especial que contém o caminho para outro arquivo. Eles podem apontar para arquivos em sistemas de arquivos diferentes ou até arquivos inexistentes. O sistema Linux utiliza *soft links* em vários locais da árvore de diretórios para simplificar a configuração do sistema. O problema com *soft links* é que cada um deles ocupa blocos do disco e uma estrutura *i-node* para representar aquele arquivo.

A substituição de *soft links* por *hard links* visa diminuir o número de *i-nodes* no sistema de arquivos, o que implica em uma redução do espaço ocupado pelas estruturas de gerência do sistema de arquivos, liberando mais espaço para arquivos. Na primeira versão do sistema, o número de *i-nodes* necessários para o sistema com os aplicativos caiu de 1941 para 1779 com a substituição dos links simbólicos, uma economia de 8,9%.

A diminuição do número de *i-nodes* visa transferir a memória utilizada para armazenar essas estruturas para armazenar dados de arquivos, uma vez que cada *i-node* ocupa 1024 bytes de espaço em disco. Isso foi feito utilizando-se o comando *mke2fs*, que permite se determinar exatamente quantos *i-nodes* devem ser criados em cada sistema de arquivos. Usando a formatação padrão do Linux, seriam criados 4000 *inodes*. Diminuindo esse valor para 1770, suficiente para armazenar todos os arquivos necessários, economizamos 2230 *i-nodes* (285440 bytes). Como o *disk-on-chip* possui 15983 kilobytes, economiz-se 1,7% do total de memória. Geralmente o *mke2fs* reserva 5% do sistema de arquivos para o super-usuário, de forma a garantir que mesmo caso o disco fique cheio, o administrador ainda possa executar programas que precisem de espaço no mesmo. No protótipo decidimos não reservar essa memória.

Mesmo com essas técnicas a economia de espaço não foi satisfatória, pois alguns dos programas e bibliotecas essenciais do sistema ocupavam espaço considerável (o maior deles era de cerca de 4 MB). Para solucionar este problema, entre algumas opções consideradas, o uso de compressão do sistema de arquivos se mostrou a opção mais eficiente.

A implementação utilizada [4] funciona da seguinte maneira: as funções *read* e *write* do sistema de arquivos do Linux (Extended Filesystem 2 [1]) são alteradas para utilizar compressão. Antes de ler um arquivo, a função *read* verifica se o arquivo possui o atributo de compressão; se não possuir, o processo de leitura ocorre como na função original. Se por outro lado o arquivo possuir o atributo de compressão então a função descomprime o conteúdo e o transfere a informação para o processo. Um procedimento equivalente ocorre quando se escreve um arquivo com atributo de compressão.

Nessa implementação o arquivo não é comprimido como um todo, mas sim dividido em blocos que são comprimidos separadamente. O tamanho desses blocos é escolhido durante a compilação do núcleo, sendo atualmente utilizado o valor de 32 kilobytes, valor máximo possível na versão atual. O algoritmo de compressão também pode ser escolhido, sendo que o protótipo atual utiliza o código Lempel-Ziv (LZ77), equivalente ao comando *gzip -9*. Ao contrário de mecanismos de compressão comuns no ambiente Windows, somente os dados do arquivo são comprimidos. Nenhuma meta informação (data de criação, alteração, atributos, etc) é comprimida, o que garante um bom desempenho para operações de manipulação. Além disso, a compressão pode ser ativada individualmente por arquivo, podendo assim ser desligada em casos onde a taxa de compressão resultante não seja significativa ou quando a velocidade de acesso se torna um fator crítico.

Os ganhos de compressão conseguidos foram de 48.5% (tamanho total dos arquivos comprimidos dividido pelo tamanho total dos arquivos não-comprimidos). Isso equivale a dizer que apesar de utilizarmos um dispositivo de armazenamento de 16 MB, conseguimos colocar até 32 MB de dados compactados nesse dispositivo. Os ganhos não são maiores porque vários arquivos já estavam compactados ou eram muito pequenos para serem compactados com eficiência: 78.5% dos arquivos possuíam tamanho inferior

a 10000 bytes. Como seria de se esperar, entretanto, o custo para acessar um arquivo aumentou, uma vez que cada acesso exige uma descompressão. No protótipo implementado com um processador K6-II 500 MHz o desempenho final percebido pelo usuário continuou bastante aceitável.

5. Trabalhos Futuros

Como trabalhos futuros, propomos a pesquisa de alguns tópicos que visam garantir uma melhor infra-estrutura ao sistema. As principais áreas de atuação do LUAR hoje se encontram no desenvolvimento de um sistema de monitoração e gerência remota, um sistema de arquivos distribuído adequado ao uso através de linhas telefônicas, uma arquitetura para a atualização de software e a montagem de laboratórios piloto para observar o impacto das máquinas em diversas configurações.

5.1. Monitoração e gerência remota

A administração de equipamentos em rede é usualmente uma tarefa onerosa, complexa e que exige pessoal especializado em cada instalação. Um dos objetivos desta pesquisa é desenvolver uma solução para monitoração e gerência remota de forma que o custo de manutenção das máquinas e das redes seja reduzido, tornando o computador popular ainda mais barato e acessível a escolas e instituições sem muitos recursos.

A expectativa do governo é que sejam instalados 300 mil computadores populares em aproximadamente 13 mil escolas por todo país nos próximos 2 anos. Isso significa que várias escolas no interior do país deverão receber computadores que deverão ser configurados, mantidos e atualizados periodicamente. Se uma solução de gerência remota não for criada, problemas podem levar à desativação de partes significativas da rede por longos períodos até que um técnico possa ser deslocado até os locais mais remotos. Com um sistema de gerenciamento de computadores, a empresa responsável pela manutenção saberá antes mesmo da escola que o computador está apresentando problemas e com a coleta de dados poderá identificar mais facilmente o problema e propor uma solução. Tais problemas poderão ser resolvidos remotamente ou com o envio de um técnico, que neste caso também será seu serviço facilitado e agilizado pelo conhecimento prévio dos problemas que irá encontrar.

Podemos descrever resumidamente uma proposta de sistema de gerência remota da seguinte forma: um servidor monitora estações em uma rede local. No servidor temos um *software* que periodicamente faz acessos às estações (clientes) para verificar se elas estão acessíveis e coletar informações. No cliente temos um outro *software* que funcionará, sem a intervenção humana, para responder as requisições do *software* servidor. Uma vez que a coleta de dados é feita regularmente, poderemos descrever o comportamento de operação das máquinas e assim detectar anomalias ou falhas.

Considerando-se os números possíveis de implantação de redes utilizando o computador popular essa arquitetura deverá ser bastante escalável para garantir que o sistema seja eficiente mesmo com centenas de milhares de máquinas sob sua supervisão. O modelo hoje sendo desenvolvido prevê uma rede hierárquica de elementos de monitoração e coleta de dados, onde cada nível é responsável pela coleta de informação de seus componentes, que serão então processadas no âmbito daquele nível, sendo que apenas um

resumo executivo contendo as principais informações do sistema são repassados aos níveis superiores. A informação não é perdida, entretanto, já que dados completos são armazenados em cada nível. Assim sendo, os níveis mais altos do sistema tem a todo momento uma visão resumida do estado do sistema, mas podem facilmente obter dados detalhados de qualquer componente ativo em qualquer nível por uma consulta direta.

Este sistema de monitoração se encontra hoje em implantação nos laboratórios pilotos que vem sendo criados, não só para verificar sua funcionalidade como sistema de monitoração do funcionamento normal do sistema, mas também como ferramentas para coleta de dados de observação sobre o comportamento dos usuários, uso de ferramentas e outros parâmetros utilizados para melhor caracterizar a utilização dos computadores.

5.2. Sistema de arquivos para linhas de baixa velocidade

Um *sistema de arquivos distribuído* (DFS - *distributed file system*) é um sistema de arquivos onde clientes, servidores e dispositivos de armazenamento estão dispersos entre as máquinas de uma rede, podendo englobar tanto segmentos de rede local quanto enlaces de uma rede de longa distância.

O desenvolvimento de um DFS envolve várias considerações de projeto, tais como espaço de nomes, operação com/sem estado, semântica de compartilhamento, métodos de acesso remoto e replicação.

Estamos pesquisando uma infra-estrutura para acesso a arquivos em larga escala através da Internet. Para tanto, está sendo concebido um DFS — temporariamente denominado KFS — com uma característica singular: ser viável para o acesso através de linhas de baixa velocidade (via modem) de um grande número de usuários. Como características desejáveis a serem estudadas citamos operabilidade, segurança, tolerância a falha e replicação.

Visando a comodidade do maior número possível de usuários, os quais podem desejar acessar seus dados a partir de uma máquina diferente da usual, é proposto o suporte a mobilidade de usuário, permitindo que o mesmo possa acessar seus arquivos a partir de qualquer máquina da rede KFS (rede de computadores usando o KFS).

Sistemas de arquivos distribuídos usuais são desenvolvidos com premissas de utilização bastante abrangentes, exigindo normalmente um grande volume de processamento de clientes e servidores para garantir requisitos como consistência em acessos concorrentes, compartilhamento de dados e recursos e outros problemas relacionados com as características de aplicações de alto desempenho com computações científicas e bancos de dados. Tais sistemas são sabidamente ineficientes/inviáveis para aplicações sobre linhas telefônicas, por não serem capazes de trabalhar com as baixas taxas de transferência e altas taxas de erros do meio de transmissão.

O que se pretende então é avaliar o perfil de utilização de arquivos remotos por um usuário em um ambiente residencial. As primeiras observações indicam que os usuários acessam documentos pessoais para visualização e atualização utilizando ferramentas como editores de texto, planilhas e formatadores de apresentação. Tais aplicações tem padrões de acesso bem definidos, com acessos sequenciais aos arquivos como um todo, criando cópias de trabalho locais, sem controle avançado de acessos concorrentes, que são extremamente raros já que documentos são normalmente alterados por apenas uma pes-

soa por vez, sendo essa pessoa normalmente o dono do mesmo. Acessos concorrentes se limitam usualmente à consulta a informações consolidadas. Com base nessas observações pretendemos verificar como o relaxamento de premissas mais abrangentes de sistemas de arquivos distribuídos convencionais pode ser utilizado para se desenvolver um sistema de arquivos eficiente para as necessidades de usuários domiciliares.

Estamos estudando os fatores que influenciam a escalabilidade de um sistema de arquivos distribuído a fim de determinar os requisitos de capacidade de clientes e principalmente para identificar as necessidades dos servidores que atenderão a grandes redes como as previstas para as escolas públicas. Acreditamos que minimizando o vínculos entre os dados dos usuários e os servidores, maximizando o número de clientes ativos suportados por um servidor, usando técnicas de diminuição de tráfego na rede (como compressão e uso de relatórios de alteração (*diffs*), por exemplo) e fazendo localmente nos clientes o máximo de processamento possível, estaremos contribuindo para uma maior escalabilidade do sistema.

Existem várias outras características desejáveis em um DFS como operabilidade, tolerância a falha, uma maior segurança, replicação, as quais, devido à complexidade de sua implantação, somente serão estudadas num segundo momento.

A maioria dos sistemas de arquivos distribuídos conhecidos utiliza *caches* como meio de aumentar o desempenho do sistema, diminuindo o tráfego de rede e o volume de entrada e saída no servidor. Baseado nessas considerações e na restrição de acesso através de linhas de baixa velocidade, sugerimos a adoção de *caches* com política de escrita postergada (*delayed-write*) para atualização de dados no servidor. Tal escolha se deve ao fato dessa política possibilitar um menor tráfego de rede e uma sensação de maior eficiência do ponto de vista do usuário.

5.3. Atualização de *software*

Com o passar do tempo, novas versões do *software* dos clientes serão lançadas tendo em vista aprimoramentos, inovações, correção de problemas e adequação a novos protocolos e padrões que se tornem difundidos pela rede. A atualização desse *software* em um sistema de grandes proporções e com uma grande variedade de usuários pode vir a ser caótica se não forem tomadas medidas no sentido de tornar essa tarefa simples para o usuário e fácil de ser manipulada pelo administrador do sistema.

Propomos o estudo de um meio de disponibilizar novas versões de *software* de modo a substituir versões anteriores, de forma gradual, evitando problemas de escalabilidade, e transparente (automática), tornando possível a gerência do sistema ainda que o mesmo assuma grandes proporções. Dessa forma o usuário não precisa ter consciência do processo de atualização e não precisará ser forçado a interromper sua operação usual para proceder a atualizações do sistema.

5.4. Implantação de laboratórios piloto

O modelo de utilização proposto para o computador popular é algo novo, sem um paralelo direto com resultados obtidos no passado para o comportamento de usuários em redes locais ou utilizando computadores pessoais usuais. Os requisitos de funcionalidade das máquinas, o padrão de uso e acesso à rede pelos usuários, o volume de tráfego gerado

pela arquitetura e pela interação entre máquinas e servidores, computadores domiciliares e servidores atrás de linhas telefônicas são hoje em grande parte uma incógnita.

Para resolver este problema o LUAR vem se unindo a diversas entidades de alcance social como escolas públicas, associações comunitárias, organizações de apoio à sociedade e às empresas, universidades e mesmo empresas particulares com interesse em soluções corporativas para criar laboratórios piloto onde as soluções propostas podem ser testadas com usuários finais em situações reais. Estes laboratórios estão sendo montados com a colaboração da indústria que começa a fornecer suas versões do conceito difundido pelo computador popular. É nosso objetivo criar uma massa significativa de usuários que nos permita determinar com mais segurança as melhores configurações para cada caso, identificar os aplicativos que melhor se adaptam às necessidades dos usuários e caracterizar a carga de processamento e o tráfego de rede que devem ser tratados por máquinas clientes, servidores e a infra-estrutura de rede em cada caso.

6. Conclusões

O computador popular é uma proposta de solução para um dos problemas que devem ser vencidos para que tenhamos uma real universalização de acesso aos serviços da Internet para toda a população. Este trabalho apresentou detalhes técnicos do protótipo desenvolvido pelo Laboratório de Universalização de Acesso do DCC-UFMG, discutindo detalhes de *hardware* e *software* envolvidos no desenvolvimento do sistema, bem como as principais decisões de projeto que afetaram a solução.

Finalmente, acreditamos que o protótipo do computador popular seja apenas um primeiro passo não só na direção de democratizar o acesso às informações e serviços através da Internet, mas como oportunidade de desenvolvermos a indústria de *software* e *hardware* nacionais, que, investindo em uma solução para países de baixo poder aquisitivo, poderiam alavancar um nicho de mercado que por enquanto é praticamente inexplorado.

Referências

- [1] Rémy Card, Theodore Ts'o, and Stephen Tweedie. Design and implementation of the second extended filesystem. *First Dutch International Symposium on Linux*, 2001. <http://e2fsprogs.sourceforge.net/ext2intro.html>.
- [2] The New Internet Computer Company. New internet computer company. <http://www.thinknic.com>, 2001.
- [3] Compaq. Home internet appliances. <http://athome.compaq.com/showroom/static/iPAQ/intappliance.asp>, 2001.
- [4] Antoine Dumesnil de Maricourt. Ext2 file system compression. <http://cvs.bofh.asn.au/e2compr>, 2001.
- [5] Eazel. Nautilus development. <http://nautilus.eazel.com/>, 2001.
- [6] Galeon-Devel. Galeon : the web, only the web. <http://galeon.sourceforge.net/>, 2001.

- [7] Be Incorporated. Beia. <http://www.be.com/products/>, 2001.
- [8] Intel. Intel web appliance. <http://www.intel.com/internetappliances/webappliance>, 2001.
- [9] KDE. K desktop environment home (kde.org). <http://www.kde.org>, 2001.
- [10] KOffice. The koffice project. <http://www.koffice.org/>, 2001.
- [11] Microsoft. Microsoft windows ce. <http://www.microsoft.com/windows/embedded/ce/default.asp>, 2001.
- [12] Sun Microsystems. Staroffice 5.2 welcome. <http://www.sun.com/staroffice/>, 2001.
- [13] Netscape. Netscape. <http://home.netscape.com/>, 2001.
- [14] Microsoft WebTV Networks. Webtv. <http://www.webtv.com>, 2001.
- [15] Linux Online. Linux online. <http://www.linux.org>, 2001.
- [16] Mozilla Organization. The mozilla organization. <http://www.mozilla.org>, 2001.
- [17] OpenOffice Source Project. www.openoffice.org. <http://www.openoffice.org/>, 2001.
- [18] Opera Software. Opera - the fastest browser on earth! <http://www.opera.com/>, 2001.
- [19] Sony. e villa network entertainment center. <http://www.evilla.com/html/index.html>, 2001.
- [20] KDE Team. Konqueror homepage. <http://www.konqueror.org>, 2001.
- [21] Trolltech. Trolltech - announce - embedded qt available for download. <http://www.trolltech.com/company/announce/embeddedd1.html>, 2001.