Integração Extended SimMan Tool & CCS – Simulação de Arquiteturas Superescalares em *Clusters*

Wagston Tassoni Staehler
Instituto de Informática – UFRGS
tassoni@inf.ufrgs.br

Guilherme Dal Pizzol
Instituto de Informática – UFRGS
gpizzol@inf.ufrgs.br

Philippe O. A. Navaux
Instituto de Informática – UFRGS
navaux@inf.ufrgs.br

Resumo

Simulação é o método mais usado e eficiente para projeto de novos processadores. Através dela podemos reproduzir e considerar os parâmetros e variáveis de uma arquitetura real, como por exemplo, arquiteturas superescalares. Com este intuito foi desenvolvido o software *de gerenciamento de simulações Extended SimMan. Esta ferramenta é uma interface gráfica para os simuladores do SimpleScalar Tool Set, gerando arquivos de configuração e gerenciando as simulações em diferentes máquinas e extraindo os dados de interesse dos arquivos de resultados e apresentando-os em gráfico e tabela.*

O Extended SimMan pode utilizar as máquinas de uma rede para executar as simulações, e também os nodos de um cluster, *através do* software *de gerenciamento de* clusters *CCS.*

A idéia, afinal, é tornar o processo de simulação o mais simples e transparente possível ao usuário final, mascarando os vários programas envolvidos bem como os formatos dos arquivos utilizados e reduzindo drasticamente o ônus no lançamento de simulações que levam muitas vezes mais de 24 horas de duração.

1. Introdução

Processadores superescalares aumentam a desempenho de processamento através da execução de instruções concorrentes [12]. Elas exploram o paralelismo entre instruções fazendo uso de *pipeline* de múltiplos estágios, múltiplas unidades funcionais, escalonamento dinâmico, mecanismos de previsão de desvios e execução fora de ordem.

Por outro lado, este aumento de desempenho torna os processadores modernos difíceis de avaliar, graças à alta complexidade dos elementos usados na sua construção. Existem três maneiras de analisar a desempenho destes prodígios da engenharia moderna: execução direta, modelos analíticos e simulação. Execução direta não é, muitas vezes, usada porque se precisa do *chip* para testar. Modelos analíticos precisos são difíceis de desenvolver e necessitam de muito conhecimento matemático.

A solução é a simulação, mais simples e eficiente como modelador e não precisa do *chip* para fazer a análise. O processador pode ser descrito em linguagem de alto nível, como C/C++, permitindo a exploração de uma grande área de projeto e facilitando a análise antes do *hardware* tornar-se disponível.

Um bom exemplo de ferramentas desenvolvidas para a simulação de processadores é o SimpleScalar Tool Set [2], um conjunto de simuladores, compiladores e utilitários que permitem desde uma simples simulação funcional até uma complexa simulação de processadores do estado-da-arte. Ferramentas estas que podem ter seu uso prejudicado devido a sua interface textual. Como o usuário deve lidar com um grande número de arquivos de configuração e de resultados, o trabalho de realizar estas ações manualmente por linha de comando torna-se oneroso em matéria de tempo e esforço.

Desta forma, uma ferramenta que possa gerenciar simulações é de grande valia para usuários de simuladores. Esta é a meta do presente trabalho: ajudar os usuários do SimpleScalar a executar simulações de uma maneira rápida e eficiente.

O Extended SimMan (*Simulation Manager*) provém as seguintes funcionalidades:

- Interface gráfica e intuitiva;
- Criação/edição de arquivos de configuração (simples/múltiplos);
- Controle de execução de simuladores em diferentes máquinas de uma rede e também com suporte a *cluster* através do gerenciador CCS;
- Visualização gráfica dos resultados (tabela, gráfico).

Para dinamizar ainda mais o processo de simulação, foi feita uma integração desta ferramenta com o CCS, de modo a tornar os nodos do *cluster* disponíveis para que o Extended SimMan execute as simulações no menor tempo possível.

2. SimpleScalar Tool Set

SimpleScalar Tool Set foi desenvolvido em Wisconsin-Madison University, com o objetivo de permitir simulações de arquiteturas usuais, como, por exemplo, as arquiteturas superescalares. É utilizado, inclusive, em centros de pesquisa de arquiteturas de computadores em todo o mundo.

Estas ferramentas implementam a arquitetura SimpleScalar, similar à arquitetura MIPS [10]. A ferramenta tem duas versões, *big-endian* e *little-endian*, para manter a portabilidade em diferentes sistemas. A semântica ISA do SimpleScalar é um superconjunto do MIPS-IV ISA, sendo chamada PISA (Portable ISA).

As principais vantagens desta ferramenta são alta flexibilidade, portabilidade, extensibilidade e desempenho. Seis simuladores orientados à execução são incluídos, os quais serão detalhados mais tarde. Além disso, a ferramenta fornece binários pré-compilados (incluindo o SPEC95) e uma versão modificada do GCC (GNU Compiler C), a qual permite a compilação de código fonte C. Um depurador e um visualizador de *pipeline* também estão disponíveis.

Simulações funcionais podem ser feitas pelo *sim-fast* e *sim-safe* onde cada instrução é executada seqüencialmente. A principal diferença entre eles é que o último verifica o alinhamento e a permissão de acesso a cada referência de memória. Simulação de memória *cache* pode ser feita facilmente com *sim-cache* e *sim-cheetah*. O segundo traz uma nova política de substituição desenvolvida pelo criador do simulador [13]. O quinto simulador disponível é chamado *sim-profile*. Ele gera o *profile* de cada classe de instruções.

O último e mais detalhado simulador é chamado *sim-outorder* e é um processador completo incluindo simulação de ciclos. Este é o simulador usado na implementação do Extended SimMan. O *sim-outorder* suporta execução fora de ordem, baseada na *Register Update Unit* (RUU) e o sistema de memória usa uma fila *load/store*. Este simulador implementa um *pipeline* de seis estágios: busca, despacho, escalonamento, execução, escrita de resultados e graduação.

Como pode ser notado, o simulador *sim-outorder* possui muitos parâmetros de configuração que podem ser alterados para obter diferentes tipos de experimentos, conforme desejado por um pesquisador de arquiteturas de computadores. Os principais parâmetros que podem ser alterados são:

- Largura dos estágios de busca, decodificação, despacho e graduação;
- Tipos de execução (*in-order*, *out-of-order*);
- Tamanho das filas de instruções (busca, despacho, *load/store*, RUU);
- Número de unidades funcionais (inteiro e ponto-flutuante);
- Configurações de *cache* (níveis unificados, associatividade, política de substituição);
- Configuração do previsor de desvios.

3. Extended SimMan Tool

Extended SimMan (*Simulation Manager*), mostrado na Figura 1, é uma ferramenta auxiliar para simulação de processadores, usando por base os simuladores do SimpleScalar Tool Set, descrito anteriormente. A ferramenta fornece três funcionalidades básicas:

- Geração de arquivos de configuração dos simuladores;
- Execução de simulações com múltiplos arquivos de configuração e *benchmarks* em várias máquinas (em uma rede ou em um *cluster*);
- Extração das estatísticas dos arquivos de resultados dos simuladores e visualização das mesmas em gráfico e tabela.

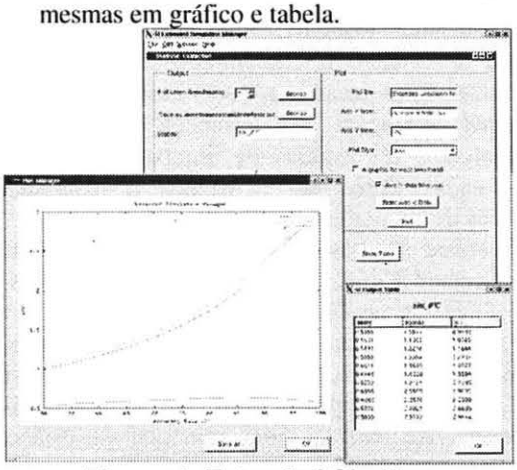

Figura 1. Extended SimMan

A ferramenta, como citada acima, permite a edição de arquivos de configuração do *sim-outorder*, o mais usado e conhecido simulador do SimpleScalar Tool Set. Este simulador, como os outros do SimpleScalar, tem sua configuração e resultados salvos em arquivos, em um formato textual próprio. Com Extended SimMan, os usuários do simulador não precisam se preocupar com estes formatos. Através de uma interface gráfica que será detalhada depois, eles poderão gerar facilmente estes arquivos. E, além disso, muitas vezes em uma avaliação de arquitetura de computador são necessários múltiplos arquivos de configuração, variando-se apenas alguns parâmetros arquiteturais. Utilizando a ferramenta, o usuário está apto a gerar vários arquivos de configuração em um único passo, escolhendo os parâmetros e os valores que mudarão de arquivo para arquivo.

Controlar a execução da simulação pode aparentemente ser uma tarefa fácil. O problema aparece quando há vários processos de simuladores em diferentes máquinas, usando diferentes arquivos de configuração e *benchmarks*. A ferramenta permite a execução do simulador em diferentes máquinas, com múltiplos arquivos de configuração e *benchmarks* conforme desejado pelo usuário. De acordo com a informação fornecida pelo usuário, a ferramenta executa os simuladores nas máquinas disponíveis, usando a correta configuração e *benchmark*. A informação provida pelo usuário inclui: *benchmarks*, configurações, máquinas e o simulador a usar. A ferramenta avisa quando a simulação inicia e quando termina.

Extended SimMan também permite que, ao final da simulação, ou com uma simulação prévia já concluída, o usuário faça a extração das estatísticas dos arquivos de resultados gerados pelos simuladores. O usuário informa a estatística a ser extraída e onde estão os arquivos. A ferramenta gera e salva um arquivo texto com estas informações. Além disso, o usuário também poderá visualizá-las em gráfico ou tabela.

Como fica evidente, qualquer pessoa que deseje utilizar os simuladores do SimpleScalar Tool Set, desde estudantes a pesquisadores, podem usar esta ferramenta. A principal vantagem do Extended SimMan é tornar transparentes ao usuário os arquivos de configuração, complicadas linhas de comandos e manipulação de arquivos de resultados.

Extended SimMan foi implementada em GNU Linux com Borland Kylix [9] e com a ajuda de scripts *bash* e *awk* [3,4]. A aplicação é MDI (*Multiple Document Interface*), i.e., a janela principal mantém todas as janelas filhas dentro de sua área cliente. Assim, o usuário pode editar múltiplos arquivos com apenas uma instância do programa. Ao mesmo tempo em que ele está editando arquivos, é possível executar e gerenciar uma simulação, como a Figura 2 mostra.

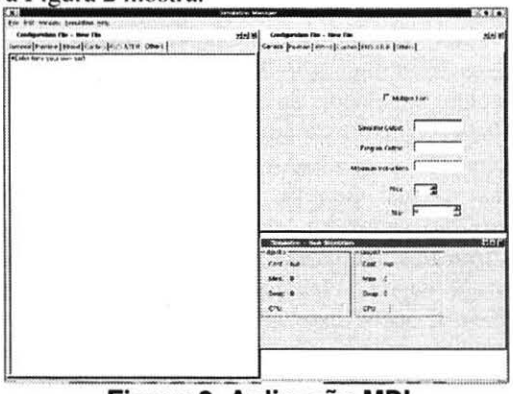

Figura 2. Aplicação MDI

Estes três módulos são independentes, ou seja, se o usuário desejar editar um arquivo de configuração ou extrair estatísticas não precisa configurar uma simulação. Eles são detalhados a seguir.

3.1 Módulo I – Edição de Arquivos de Configuração

O primeiro módulo é o gerador de arquivos de configuração para o simulador *sim-outorder*. Neste módulo, o usuário pode gerar, de uma forma rápida, fácil e eficiente, um ou múltiplos arquivos de uma vez. Na verdade, ele não precisará conhecer nem o formato de um arquivo de configuração. A ferramenta cuida disto.

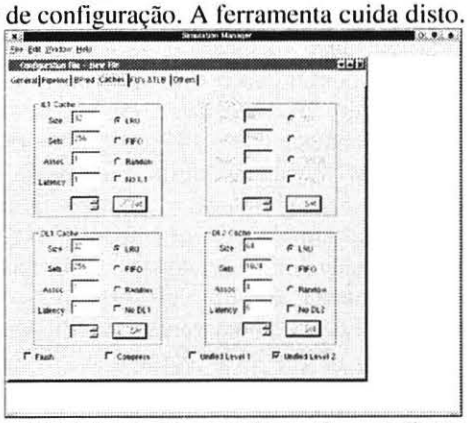

Figura 3. Edição de arquivos de configuração

Este módulo está dividido em categorias de configuração, i.e., o usuário tem as configurações separadas em classes. Esta separação pode ser vista diretamente na interface, já que ela foi implementada como uma janela com múltiplas páginas, como pode ser visto na Figura 3.

As seguintes classes de configuração são oferecidas:

- **General:** parâmetros específicos do simulador são encontrados aqui. Por exemplo, arquivos de saída de *benchmarks* e simuladores e números de instruções a serem executadas. A opção que habilita a criação de múltiplos arquivos também está aqui.
- **Pipeline:** as opções específicas do *pipeline* do *sim-outorder* são encontradas nesta página. Larguras de busca, decodificação, despacho ou de graduação podem ser configuradas aqui.
- **Bpred:** a configuração do previsor de desvios é encontrada aqui. Um dos seis diferentes previsores de desvio implementados pelo *sim-outorder* pode ser escolhido aqui. Dependendo do previsor escolhido, configurações específicas são habilitadas, tornando a tarefa ainda mais fácil para usuários que não estão habituados com o processo.
- **Caches:** nível 1 e 2 podem ser configuradas aqui. É possível definir se cada nível será unificado ou não e suas latências específicas. Além disso, para cada *cache* é possível definir o número de

conjuntos, associatividade, número de linhas e blocos, política de substituição.

- **FU's & TLB:** nesta página estão as configurações das TLB's, memória principal e unidades funcionais.
- **Others:** qualquer outra configuração pode ser inserida aqui, bem como comentários. Esta página funciona como um pequeno editor de texto. Ela pode ser usada para gerar arquivos de configuração para um simulador que use o *sim-outorder* como base. Por exemplo, o simulador ss-smt de arquiteturas SMT, desenvolvido por Gonçalves [6].

Numa simulação, o usuário geralmente deseja testar sua arquitetura alvo com diferentes valores para diversos parâmetros (por exemplo, combinando previsores de desvio com diferentes configurações de *cache*). Isto implica em vários arquivos de configuração, cada um diferindo de apenas uma ou duas linhas/colunas do outro. Gerar manualmente estes arquivos não é muito simples, já que cada arquivo deve ser editado, procurando pela linha/coluna que deve ser mudada. Esta ferramenta minimiza este problema. O usuário somente informa os parâmetros que devem ser alterados com seus respectivos valores, e a ferramenta gera os arquivos, não importando o número deles.

O comportamento padrão do módulo/ferramenta é gerar apenas um arquivo com os valores inseridos/modificados pelo usuário. Para ativar a criação de múltiplos arquivos, é necessário marcar a *checkbox* na página "General". Desta forma, o modo múltiplos arquivos está ativado.

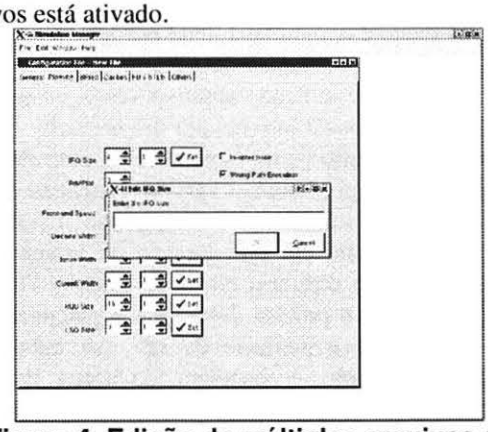

Figura 4. Edição de múltiplos arquivos de configuração

O modo múltiplos arquivos apresenta caixas com o número de valores a serem inseridos para cada parâmetro, bem como um botão "Set", que deve ser clicado para adicionar os valores. Quando a configuração é salva, a ferramenta gera um nome padrão, criado a partir do nome do parâmetro. Por exemplo, se o usuário mudar três vezes os valores do parâmetro X e duas vezes o parâmetro Y, a ferramenta gerará arquivos com os respectivos nomes: ConfX1_Y1, ConfX1_Y2, ConfX2_Y1, ConfX2_Y2, ConfX3_Y1, ConfX3_Y2. Este modo pode ser visto na Figura 4.

3.2. Módulo II – Controle de Simulações

Este é o principal módulo da aplicação. Ele é responsável por definir, executar e controlar simulações. Com este módulo, o usuário não precisará gerenciar manualmente a simulação, i.e., ele não precisará acessar várias máquinas para disparar seus processos, e além disso, não precisará se preocupar com o fim de uma execução para disparar outro processo na mesma máquina.

Com a ferramenta Extended SimMan, através deste módulo, o usuário apenas define as máquinas (Figura 5), *benchmarks* e arquivos de configuração que serão usados na simulação. Com estas informações, a ferramenta começa a executar os simuladores nas dadas máquinas. De tempos em tempos (o tempo pode também ser definido), a ferramenta checa o estado da execução em cada máquina e dispara novos processos quando possível. Quando não existem mais processos rodando, a ferramenta finaliza a simulação e avisa o usuário por e-mail e também na tela da aplicação. Todos os resultados estarão no diretório informado pelo usuário no início da simulação. E ainda, o usuário pode salvar a configuração da simulação (dados como arquivos envolvidos, máquinas) para uso posterior em outras simulações.

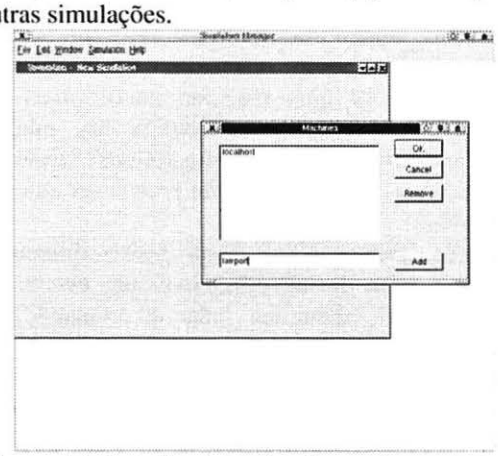

Figura 5. Adição de máquinas de uma rede

O usuário tem duas opções para iniciar a simulação. Ele pode começar uma simulação desde o início ou partir de uma simulação salva previamente. No caso de uma simulação nova, o usuário deve informar se as máquinas disponíveis são da rede ou de um *cluster*, afinal elas possuem tratamentos diferentes. As máquinas da rede são acessadas via comandos remotos ssh, e as máquinas do *cluster* são acessadas através do *software* de gerenciamento de *clusters* CCS [8]. Esteja claro que o usuário não toma conhecimento da existência deste

software, pois o Extended SimMan mascara o seu funcionamento.

Por intermédio do CCS, o Extended SimMan informa ao usuário quantos nós estão disponíveis, e este indica quantas máquinas lhe são necessárias. Após alocar o número de nós desejados, a conexão via rsh somente é habilitada ao usuário que os alocou, e assim o Extended SimMan pode disparar os processos dos simuladores.

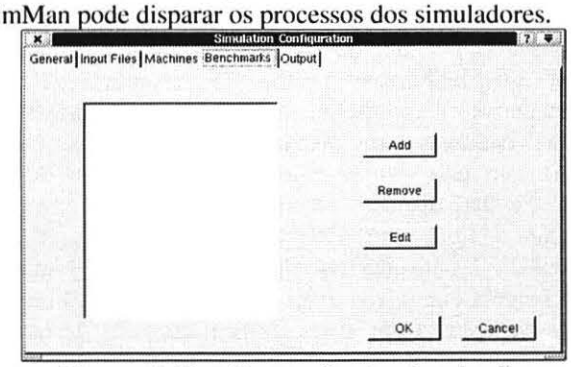

Figura 6. Configuração da simulação

Para iniciar uma simulação, além de máquinas, o usuário deve informar os *benchmarks* e arquivos de configuração. Para tanto, a configuração da simulação precisa ser acessada, através do menu "Simulation". Esta configuração é dividida em seções: General, Machines, Benchmarks e Configurations (Figure 6).

Na seção "Machines", máquinas podem ser adicionadas ou removidas. Nomes válidos de máquinas precisam ser usados ou endereços IP. Não há nenhum atributo especial para máquinas duais (com dois processadores). Se uma máquina possui mais de um processador, é necessário adicioná-la este número de vezes. Também na seção "Configurations", arquivos de configuração usados na simulação podem ser adicionados ou removidos.

A seção "Benchmarks" é um pouco diferente das outras, pois além de inserir/remover um *benchmark* da lista, é possível editar sua linha de comando. Isto é permitido porque um *benchmark*, como uma aplicação normal, pode ter parâmetros por linha de comando. Um exemplo é o *benchmark* gzip, que tem os parâmetros: -9 <input file>. Então, se os *benchmarks* têm parâmetros, eles podem ser adicionados pela sua edição, depois do caminho + nome do *benchmark*.

Configurações opcionais podem ser encontradas na seção "General", onde algumas opções globais podem ser alteradas: o diretório dos resultados (o padrão é home do usuário), o simulador a ser executado (o padrão é *sim-outorder*) e o tempo de *refresh* (o padrão é 30s). O tempo de *refresh* é o intervalo usado para atualizar as informações de cada processo que está executando. Além de atualizar as informações na tela, novos processos podem ser disparados se houver uma máquina disponível.

A configuração de simulações a serem rodadas no *cluster* é similar, mudando apenas a seção "Machines". Em vez disso, a ferramenta oferece o número de nodos disponíveis para que o usuário informe a quantidade de que necessitará e por quanto tempo.

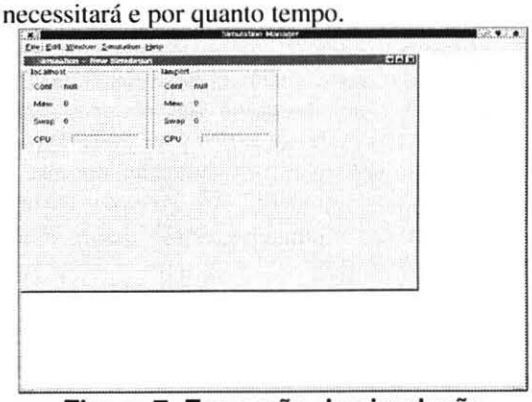

Figura 7. Execução da simulação

Com todo este conjunto de configurações, o processo de simulação pode ser iniciado. A partir daqui, o usuário pode acompanhar o estado das simulações durante a execução dos processos. O usuário pode verificar, pela tela, a memória física disponível e memória *swap*. Além disso, o usuário pode ver a quantidade de CPU utilizada. Todas estas informações são atualizadas de acordo com o tempo de *refresh* previamente setado pelo usuário. Esta tela pode ser vista na Figura 7.

A conexão com máquinas remotas, bem como a extração de informações delas, é feita por escripts *bash* + *awk*, que usam ssh e/ou rsh para conectar nas máquinas remotas. Por padrão, toda a comunicação entre a máquina local e as máquinas remotas é feita por ssh, por causa dos bem conhecidos problemas de segurança do rsh. Apesar disso, rsh pode ser utilizado algumas vezes, como no caso de *clusters* isolados da Internet por um *firewall*.

O comportamento padrão de ssh/rsh é usar um método de autenticação por senha. Para o Extended SimMan trabalhar apropriadamente, é necessário desabilitar a checagem de senha em rsh, através da criação de um arquivo .rhosts no diretório home do usuário. Do mesmo modo, ssh também precisa desabilitar a autenticação por senha só que diferentemente do rsh, ssh trabalha com criptografia, usando o conceito de troca de chaves pública/privada. Logo, o usuário precisa criar o par de chaves pública/privada para desabilitar a autenticação por senha.

Como dito antes, a conexão e extração de dados são realizadas utilizando scripts *bash* que usam ssh/rsh. Isto é possível graças ao conceito de *pipe*, uma canalização onde os dados salvos em um processo são colocados na entrada de outro processo, por exemplo.

3.3 Módulo III – Extração e Visualização de Estatísticas

Em um experimento razoavelmente simples, por exemplo, com oito *benchmarks* e dez arquivos de configuração (dez valores diferentes para o mesmo parâmetro), serão feitas oitenta simulações gerando oitenta arquivos de saída. Então, o usuário que desejar extrair uma simples estatística destes arquivos precisará abrir oitenta arquivos, procurar pela estatística e extrair seu valor. Uma tarefa relativamente árdua.

Com a ajuda do Extended SimMan, através do terceiro módulo, este trabalho é simplificado. O usuário deve definir onde estão os arquivos de saída e o arquivo onde o Extended SimMan deverá escrever os resultados. Aqui, o usuário deverá informar à ferramenta a estatística a ser procurada e extraída no formato em que ela é encontrada no arquivo de saída do *sim-outorder*. Por exemplo, se o usuário desejar extrair a estatística IPC (Instruções por Ciclo) deverá informar a ferramenta que ela deve extrair a estatística sim_IPC. Para auxiliar o usuário que não está familiarizado com este formato, existe uma tabela com estas informações presente no "Help" da ferramenta. Este módulo pode ser visto na Figura 8.

Ao informar quais arquivos contêm os resultados das simulações, o usuário deve definir uma ordem (ascendente/descendente) dos arquivos. Isto é necessário porque a ferramenta não está apta a saber quais arquivos correspondem a quais pares de *benchmarks*-configurações.

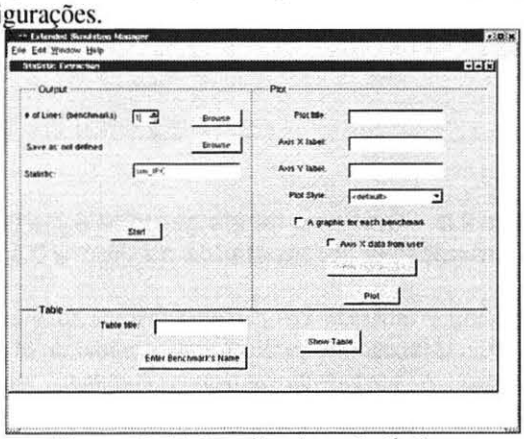

Figura 8. Extração de estatísticas

O usuário ainda deve informar o número de *benchmarks*, assim a ferramenta irá abrir tantas janelas quanto o número informado, e o usuário deverá selecionar todos os arquivos relativos a cada *benchmark*, em ordem ascendente/descendente da variação do parâmetro modificado.

Na simulação com 2 (dois) *benchmarks* (bench1, bench2) e 4 (quatro) arquivos de configuração (conf1, conf2, conf3, conf4), o usuário deve informar 2 (dois) como o número de linhas. Clicando em "Browse", a tela de seleção aparecerá duas vezes. Na primeira vez, o usuário deverá selecionar os arquivos que correspondem as seguintes simulações: Simul_Conf1_Bench1, Simul_Conf2_Bench1, Simul_Conf3_Bench1, Simul_Conf4_Bench1. Na segunda vez, o usuário deve repetir o mesmo procedimento, mudando Bench1 por Bench2. Quando tudo estiver feito, ele permite escolher o arquivo de destino e a estatística a ser extraída.

Depois disso, o usuário pode visualizar os resultados em gráfico ou tabela. E ainda, a tabela pode ser salva no formato CSV e o gráfico pode ser salvo como um arquivo PNG, para futura inserção em um relatório de pesquisa.

Para apresentar o gráfico, o Extended SimMan usa o Gnuplot [5]. A ferramenta permite configurar todos os parâmetros do Gnuplot, como: tipo do gráfico, título e nomes dos eixos. O usuário pode, inclusive, escolher se todas as curvas estarão em único gráfico ou se cada curva estará em um gráfico.

4. CCS

CCS (*Computing Center Software*) [8] é um *software* através do qual é feita a submissão de trabalhos do *cluster*. Seu objetivo é proporcionar a possibilidade de qualquer usuário, com permissão, alocar e utilizar nodos do *cluster* para suas aplicações. O usuário deve informar o número de nodos que deseja e por quanto tempo deseja reservar estas máquinas. O CCS verifica se a solicitação é válida, e em caso positivo, retorna o número da reserva.

CCS, ao realizar as reservas, aloca as máquinas do *cluster* por nodo e não por processador. Isto é uma característica boa para aplicações distribuídas, mas não muito para a execução de simulações. Porém, o *cluster* utilizado atende realmente um grande número de aplicações distribuídas, sendo preferido o uso do CCS. Isto não se tornou um problema para a integração, já que o Extended SimMan executa, por exemplo, dois processos de cada vez quando uma máquina é bi-processada, fazendo um bom aproveitamento do *hardware* disponível.

A partir do momento em que o CCS alocou as máquinas, fica autorizada a conexão com os nodos por rsh apenas para o usuário requisitante.

O CCS mantém arquivos com informações atualizadas do que está se passando no *cluster*. É deles que o Extended SimMan obtém os dados de que necessita para auxiliar o usuário nas decisões.

Os principais comandos CCS, que foram utilizados nesta ferramenta são:

- *ccsalloc*: serve para requisitar nodos do *cluster*. Devolve o número da reserva.
- *ccsinfo*: apresenta os nomes das máquinas que foram fornecidas para atender à requisição. Possui uma opção para ver a fila do *cluster*, com todos os usuários logados, quantos nodos

utilizados por cada um e quanto tempo de uso ainda resta.

- *ccsrun*: outra forma de executar um comando remoto no nodo.

5. Integração Extended SimMan - CCS

O Extended SimMan facilita o trabalho do pesquisador de arquiteturas de computadores, mantendo-o alheio ao tratamento de todos os arquivos envolvidos. Também verifica o andamento das execuções dos processos nas diversas máquinas para poder disparar novos. Além disso, um outro fator relevante que motivou este trabalho é o tempo que estas simulações levam. O ideal é utilizar o maior número de máquinas possível quando se tem um experimento envolvendo muitas simulações. Uma solução é a utilização dos nodos do *cluster*.

É uma opção a mais para o usuário executar suas simulações, e uma boa solução no que diz respeito à duração do experimento, caso vários nodos estejam livres.

Ao nível de implementação, o que o Extended SimMan faz é acessar os arquivos com os dados atualizados do *cluster* e oferecer ao usuário as opções de máquinas e tempos disponíveis.

O que o usuário deve fazer é selecionar os arquivos de configuração e *benchmarks*, assim como um diretório para depositar os arquivos de resultados, e deve também indicar quantos dos nodos disponíveis deseja utilizar. Depois é só aguardar enquanto a ferramenta segue disparando as simulações até ter concluído todas.

6. Validação da Ferramenta

Para validar a ferramenta, atestando que todas funcionalidades estão trabalhando como esperado, alguns testes foram feitos. Como a ferramenta foi desenvolvida em módulos, cada módulo teve seu próprio teste. Alguns resultados de um módulo foram usados como ponto de partida para os seguintes.

No módulo I, alguns arquivos de configuração foram gerados. Todos eles foram testados com o simulador *sim-outorder*, sendo submetidos à checagem sintática. Todas as combinações de possibilidades entre configurações de *caches* e previsores de desvio foram testadas, e todas passaram pela checagem sintática feita pelo *sim-outorder*. Além disso, múltiplos arquivos foram gerados, analogamente a uma simulação real, sendo usados na validação dos outros módulos.

Para fazer a validação do segundo módulo, foi definida uma simulação similar a uma que já havia sido executada em [11]. Para realizar isto, os mesmos *benchmarks* (seis) e as mesmas configurações (onze, geradas pelo módulo I) foram usados. Nesta simulação oito máquinas foram usadas, onde quatro delas são biprocessadas e foram usadas com conexões rsh (Dual Pentium Pro *Cluster*); no restante foram usadas conexões ssh e elas eram apenas estações de trabalho com Pentium III. Todas as máquinas tinham GNU/Linux como sistema operacional. Ambas as conexões rsh e ssh foram feitas com sucesso sem nenhum problema decorrente delas. Os arquivos de resultados das simulações foram muito similares àqueles da simulação anterior (os resultados do *sim-outorder* podem ser diferentes quando é usada uma máquina local diferente).

Finalmente, no módulo III, as estatísticas que haviam sido retiradas manualmente no trabalho citado acima, foram, agora, extraídas com o auxílio da ferramenta. A extração permitiu a geração dos mesmos gráficos, como era esperado. Na Figura 9.a, o formato do arquivo de saída pode ser visto. O usuário, neste caso, informou que o número de linhas deveria ser seis, representando o número de *benchmarks*, ficando o número de colunas igual a onze, que é número de arquivos de configuração (como explicado na seção anterior). Na Figura 9.b os respectivos gráfico e tabela podem ser vistos.

```
0.5368 0.5531 0.5697 0.5858 0.6011 0.6148 0.6253 0.6305 0.6269 0.6076 0.5639
1.0574 1.1361 1.2274 1.3354 1.4648 1.6220 1.8161 2.0568 2.3574 2.7021 2.9772
0.9990 1.0745 1.1644 1.2701 1.4027 1.5584 1.7395 1.9695 2.2908 2.6695 2.9814
1.2072 1.2445 1.2934 1.3471 1.4049 1.4701 1.5609 1.6530 1.7770 1.9177 2.0605
1.0837 1.1578 1.2429 1.3425 1.4587 1.5970 1.7639 1.9668 2.2142 2.4905 2.7185
0.8604 0.9353 1.0245 1.1330 1.2680 1.4399 1.6667 1.9794 2.4371 3.1290 3.8667
```

Figura 9.a – Saída gerada pelo módulo III

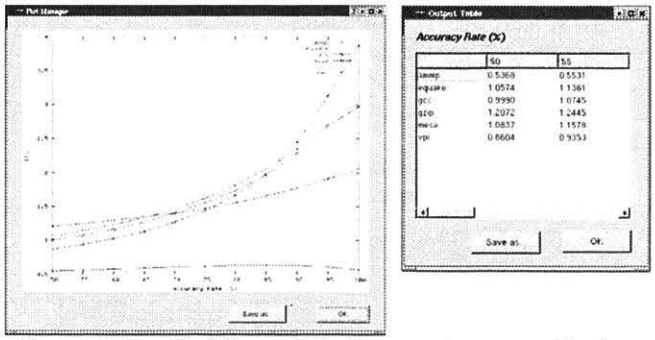

Figura 9.b – Gráfico e tabela gerados a partir dos resultados apresentados na Figura 9.a

Quando é utilizado um *cluster* e não as máquinas de uma rede, o procedimento é o mesmo depois de alocados os nodos. A ferramenta utiliza o comando rsh para atualizar e disparar novos processos exatamente como no outro caso.

7. Conclusões

Neste artigo, foi apresentado Extended SimMan integrado com CCS. Esta ferramenta de auxílio à simulação é um *front-end* gráfico para os simuladores do SimpleScalar Tool Set, permitindo a gerência da configuração e execução das simulações, bem como a geração do gráfico dos resultados.

Como dito antes, a tarefa da simulação não é trivial. Em qualquer análise de arquitetura, há muitas variáveis a

serem alteradas, refletindo em vários arquivos de configuração desta arquitetura específica. Além disso, é necessário analisar a arquitetura sob diferentes cargas de trabalho. Isto se reflete no uso de diversos *benchmarks*. Por exemplo, num estudo que usa doze diferentes arquivos de configuração e oito *benchmarks* distintos, 96 (noventa e seis) execuções se farão necessárias. É preciso disparar e controlar processos em diversas máquinas para reduzir o tempo de simulação. Se houver três máquinas disponíveis, o usuário precisa acessá-las e disparar os processos um a um. O usuário também poderia disparar mais de um processo por máquina, o que tornaria a máquina muito lenta, já que os simuladores são CPU *bound*. Logo, o usuário precisa monitorar estes processos para ter certeza de que a simulação prévia está terminada, para então disparar uma outra. Outro grande problema é a extração de estatísticas. Neste caso, são 96 arquivos de resultados que devem ser abertos e analisados em busca da estatística desejada.

Através da ferramenta apresentada neste trabalho, este processo é automatizado. Ela encobre vários outros *softwares* necessários à realização e análise das simulações (*sim-outorder*, CCS, Gnuplot), deixando esta parte trabalhosa e custosa em questão de tempo e esforço transparente ao usuário. O usuário somente precisa configurar sua simulação e a ferramenta faz todos os testes e execuções, informando quando terminar. Para extrair estatísticas, ele precisa conhecer apenas quais arquivos contêm as informações dos resultados e onde eles estão localizados. Com estes dados, a ferramenta extrai as estatísticas, estando preparada para exibir os respectivos gráfico e tabela, e salvá-los.

Os simuladores SimpleScalar são empregados por pesquisadores de todo mundo no projeto de arquitetura de processadores. Extended SimMan vem facilitar o trabalho destes e de alunos, acelerando o processo das medidas e testes de desempenho de novas arquiteturas. Os pesquisadores, perdendo menos tempo com o processo de simulação, poderão voltar seus esforços numa maior análise dos resultados.

8. Referências

[1] Austin, T. M. A User's and Hacker's Guide to the SimpleScalar Architectural Research Tool Set. Intel MicroComputer Research Labs. January 1997.

[2] Burger, D.; Austin, T. M. The SimpleScalar Tool Set: Version 2.0. Madison: University of Wisconsin, 1997. (Technical Report, n.1342).

[3] Free Software Foundation – Gawk. http://www.gnu.org/software/gawk/

[4] Free Software Foundation – Bash. http://www.gnu.org/software/bash/

[5] GnuPlot - http://www.ucc.ie/gnuplot/gnuplot.html

[6] Gonçalves, R. A. L.; Ayguadé, E.; Valero, M.; Navaux, P.O.A. A Simulator for SMT architectures: Evaluating Instruction Cache Topologies. XII SBAC-PAD, October 2000.

[7] Johnson, M. Superscalar Microprocessor Design. Englewood Cliffs, New Jersey: Prentice Hall, 1991. 288p. Series in Innovative Technology.

[8] Keller, A. OpenCCS Administrator Manual. Alemanha: Paderborn Center for Parallel Computing; v0.8, 114p, Outubro, 2002.

[9] Kylix – Borland. http://www.borland.com/kylix/index.html

[10] MIPS. R10000 Microprocessor User's Manual: Version 1.0. Mountain View, California: MIPS Technologies, 1995.

[11] Pizzol, G. D.; Pilla, M. L.; Navaux, P. O. A. Branch Prediction X Performance: An Analisys on SuperScalar Processors. IN: XIII SBAC-PAD. October 2001.

[12] Smith, J.E.; Sohi, G.S. The Microarchitecture of SuperScalar Processors. Proceedings of the IEEE, [S.l.], v.83, n.12, Dec. 1995.

[13] Sugumar, R. A.; Abraham, S. G. Efficient Simulation of Caches under Optimal Replacement with Applications to Miss Characterization. IN: ACM SIGMETRICS CONFERENCE ON MEASUREMENT AND MODELING OF COMPUTER SYSTEMS. Proceedings.. . pp. 24-35, May 1993.