

Boletim Técnico da Escola Politécnica da USP
Departamento de Engenharia de Computação e
Sistemas Digitais

ISSN 1413-215X

BT/PCS/0514

Algoritmo Adaptativo de
Substituição de Páginas LRU-WAR

Hugo Henrique Cassettari
Edson Toshimi Midorikawa

São Paulo - 2005

O presente trabalho é parte da dissertação mestrado, apresentada por Hugo Henrique Cassettari, sob orientação do Prof. Dr. Edson Toshimi Midorikawa: "Análise da Localidade de Programas e Desenvolvimento de Algoritmos Adaptativos para Substituição de Páginas", defendida em 26/01/2004, na EPUSP.

A íntegra da dissertação encontra-se à disposição com o autor e na Biblioteca de Engenharia de Eletricidade da Escola Politécnica da USP.

FICHA CATALOGRÁFICA

Cassettari, Hugo Henrique

Algoritmo adaptativo de substituição de páginas LRU-WAR
/ Hugo Henrique Cassettari, Edson Toshimi Midorikawa. -- São Paulo : EPUSP, 2005.

13 p. -- (Boletim Técnico da Escola Politécnica da USP, Departamento de Engenharia de Computação e Sistemas Digitais ; BT/PCS/0514)

1. Gerência de memória 2. Sistemas operacionais 3. Engenharia de computação I. Midorikawa, Edson Toshimi II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais III. Título IV. Série
ISSN 1413-215X

ALGORITMO ADAPTATIVO DE SUBSTITUIÇÃO DE PÁGINAS LRU-WAR

HUGO HENRIQUE CASSETTARI
EDSON TOSHIMI MIDORIKAWA

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia de Computação e Sistemas Digitais
São Paulo-SP, 05508-900
{hugo.cassettari,edson.midorikawa}@poli.usp.br

RESUMO

Algoritmos de substituição de páginas influenciam diretamente o desempenho de sistemas computacionais que utilizam memória virtual paginada. Algoritmos adaptativos têm a capacidade de modificar o seu comportamento ao longo do tempo, de acordo com as características de processamento observadas. Este artigo apresenta um novo algoritmo adaptativo para substituição de páginas, cujo objetivo é minimizar falhas detectadas no algoritmo LRU sem perder a sua simplicidade computacional: o LRU-WAR (*LRU with Working Area Restriction* / LRU com Confinamento da Área de Trabalho).

ABSTRACT

Page replacement algorithms have direct influence on the performance of computing systems which use paged virtual memory. Adaptive algorithms are capable of modifying their own behavior through time, depending on the execution characteristics. This paper presents a new adaptive algorithm for page replacement: LRU with Working Area Restriction (LRU-WAR), whose target is to minimize failures detected in LRU algorithm, preserving its low overhead.

1. INTRODUÇÃO

Um bom gerenciamento de memória é característica indispensável para um sistema operacional executar satisfatoriamente tarefas que exigem alto desempenho (MIDORIKAWA, 1997). Um dos aspectos mais importantes em um sistema de memória virtual paginada é a política de substituição de páginas utilizada. Esta política define os critérios de seleção empregados para a retirada de uma ou mais páginas da memória principal quando uma outra precisa ser alocada e não há espaço disponível.

O esquema de substituição de páginas mais utilizado em sistemas operacionais modernos é aquele que implementa alguma forma de aproximação do algoritmo LRU (*Least Recently Used* – que substitui a página residente acessada há mais tempo). Esta política se mostra relativamente eficiente na grande maioria dos casos, em comparação com outras políticas tradicionais como FIFO (*First In, First Out* – página carregada há mais tempo), MRU (*Most Recently Used* – página acessada há menos tempo), LFU (*Least Frequently Used* – página que recebeu menos acessos), etc.

Isto porque os programas normalmente não apresentam padrões de acesso bem definidos, ou seja, é difícil prever que página será ou não acessada brevemente. Nessa situação, o emprego do critério LRU tende a ser o mais lógico dentre as políticas de substituição triviais. Afinal, se não há mecanismos suficientes para se prever qual página levará mais tempo para ser requisitada pelo processador, substitui-se, em caso de falta, a página que está há mais tempo sem ser acessada, aquela menos ativa no momento da execução.

Porém, Jiang; Zhang (2002) apontam situações em que certamente o algoritmo LRU apresenta deficiências:

- Acessos sequenciais em um grande número de páginas de memória distintas;
- Acessos dentro de grandes *loops* (muitas páginas referenciadas por iteração);
- Frequência irregular de acessos a uma mesma página.

Esta terceira situação não apresenta a mesma inércia comportamental quanto ao padrão de acessos que as duas primeiras. A tentativa de prever variações na frequência de acessos às páginas exige estruturas adicionais que armazenem informações históricas relevantes no decorrer da execução. A alta complexidade de implementação inerente a essa tentativa (*overhead*) torna-se um grande problema e pode torná-la desinteressante.

Por outro lado, as duas primeiras situações – acessos sequenciais e acessos dentro de grandes *loops* – podem ser facilmente detectadas em um processo, pois possuem uma característica extremamente bem definida: muitas faltas de página em um curto intervalo de tempo. Outra característica fundamental é que as páginas carregadas deixam de ser referenciadas rapidamente, sem que haja uma reutilização significativa enquanto permanecem residentes na memória.

O conceito de adaptatividade em algoritmos de substituição de páginas pode trazer benefícios importantes, como a correção de deficiências presentes nas políticas tradicionais – ou triviais – e a possibilidade de otimização nos critérios de substituição adotados. Após um amplo estudo teórico nas propriedades do modelo LRU, chegamos à proposta de um novo algoritmo adaptativo: o LRU-WAR (*LRU with Working Area Restriction* / LRU com Confinamento da Área de Trabalho), cujo objetivo é minimizar as falhas detectadas no algoritmo LRU sem sobrecarregar o sistema de gerenciamento de memória.

A Seção 2 deste artigo relata alguns algoritmos adaptativos de substituição de páginas importantes, recentemente publicados na literatura, enquanto a Seção 3 detalha o novo algoritmo LRU-WAR. Os resultados de desempenho obtidos em simulações realizadas com os algoritmos LRU e LRU-WAR são exibidos e analisados na Seção 4. A conclusão do trabalho, por fim, é exposta na Seção 5.

2. TRABALHOS RELACIONADOS

Várias propostas de algoritmos adaptativos para substituição de páginas foram apresentadas à comunidade científica nos últimos anos, a maioria tendo como ponto de partida o tradicional e relativamente eficiente algoritmo LRU. Nas condições em que ele apresenta bom desempenho, sua atuação original é mantida; nas demais condições, uma alternativa de comportamento é desenvolvida. Entre as propostas nesta linha se enquadram os algoritmos SEQ – que identifica acessos sequenciais procurando adjacências entre as páginas recentemente carregadas na memória – e EELRU – o qual detecta padrões sequenciais analisando a concentração de acessos recentes em páginas que já deixaram a memória.

O algoritmo SEQ (GLASS; CAO, 1997) pode ser considerado uma versão adaptativa do LRU que procura corrigir a perda de desempenho ocasionada pela presença de acessos linearmente seqüenciais. De acordo com os padrões de acesso apresentados pelo programa em execução, o algoritmo mantém o critério original LRU ou, quando identifica um ou mais conjuntos de referências a endereços de memória seqüencialmente adjacentes, implementa um critério de substituição pseudo-MRU.

O algoritmo EELRU, *Early Eviction LRU* (SMARAGDAKIS; KAPLAN; WILSON, 1999), foi proposto como uma outra tentativa de mesclar o LRU e o MRU, baseado apenas nas posições da fila LRU em que as referências à memória se concentram. Esta fila nada mais é do que a própria representação da memória principal no modelo LRU, ordenada de forma decrescente sob o critério da recência de acesso das páginas residentes. O algoritmo EELRU detecta potencialmente qualquer tipo de acesso seqüencial, mesmo que eles não ocorram em endereços contíguos de memória.

Outro algoritmo importante é o LIRS (JIANG; ZHANG, 2002) – *Low Inter-reference Recency Set*. Seu objetivo é minimizar as deficiências apresentadas pelo LRU utilizando um critério adicional interessante: a chamada IRR (*Inter-Reference Recency*), que representa o número de páginas referenciadas entre os dois últimos acessos consecutivos a uma mesma página. O algoritmo pressupõe uma inércia comportamental e, de acordo com as IRRs coletadas, substitui a página que provavelmente levará mais tempo para ser novamente acessada. Isto significa que o LIRS não substitui necessariamente a página referenciada há mais tempo, mas ele utiliza o histórico recente desta informação para prever quais páginas têm maiores probabilidades de acesso em um futuro breve.

Três outros algoritmos trouxeram a proposta de identificar certas características de execução nos processos e se adaptar dinamicamente a elas, alternando o critério de substituição de acordo com o padrão de acessos constatado. As três criações são muito parecidas e seu objetivo comum é empregar sempre um critério ótimo de substituição para o padrão de acessos vigente na execução. Estes algoritmos são:

- DEAR (CHOI et al., 1999) – *DEtection-based Adaptive Replacement*;
- AFC (CHOI et al., 2000) – *Application/File-level Characterization*;
- UBM (KIM et al., 2000) – *Unified Buffer Management*.

As propostas aqui relatadas, entre outras, agregam benefícios conceituais importantes aos algoritmos tradicionais nos quais se baseiam, mas todas elas também possuem uma complexidade muito maior de implementação. Por outro lado, o algoritmo adaptativo LRU-WAR se destaca por detectar e tratar eficientemente períodos com predominância de acessos seqüenciais (a principal falha da política LRU, base de seu desenvolvimento), mantendo praticamente a mesma complexidade de implementação que o algoritmo LRU original, um grande diferencial positivo.

3. O ALGORITMO LRU-WAR

As políticas de substituição de páginas mais eficientes tentam explorar, cada uma ao seu modo, as características de localidade de referências inerentes aos programas (DENNING, 1980), sobretudo localidade temporal. O algoritmo LRU, por exemplo, procura substituir páginas que não tenham sido recentemente referenciadas, ou seja, páginas que não apresentem a tendência de acessos contínuos no momento de execução observado. Entretanto, como foi ponderado na Seção 1, este algoritmo possui algumas deficiências, principalmente quando o programa em execução exhibe padrões de

acesso basicamente seqüenciais – e, conseqüentemente, baixa localidade temporal em muitas páginas de memória –, situação que acontece quando uma estrutura de dados (por exemplo vetor, matriz, lista ligada, árvore) é percorrida, elemento a elemento, por um longo intervalo de acessos. Isto pode ocorrer por diversos motivos: iniciação ou atualização de elementos, busca seqüencial, processamento consecutivo – como multiplicação de matrizes –, entre outros.

Uma nova alternativa adaptativa para solucionar o mau desempenho do algoritmo LRU na presença de acessos com tais características é aqui apresentada: o algoritmo LRU-WAR (*LRU with Working Area Restriction – LRU com Confinamento da Área de Trabalho*). Baseado no tamanho máximo do *working set* estimado e validado entre duas faltas de página consecutivas, e na variação deste tamanho entre as faltas seguintes, o algoritmo utiliza um mecanismo ainda inexplorado pelas técnicas atuais: a dimensão máxima do *working set* temporário como fator decisivo de adaptatividade.

3.1. Idéia Geral

Working set (conjunto de trabalho), conforme definido por Denning (1968), pode ser descrito como o conjunto das páginas acessadas pelo programa em um determinado intervalo de tempo. Subentende-se aqui, como *working set* temporário, o conjunto das páginas acessadas entre as duas últimas faltas de página ocorridas, qualquer que seja o instante do processamento. Em outras palavras: é o conjunto das páginas recentemente utilizadas pelo programa. Um novo conceito pode então ser proposto: definimos *área de trabalho* (*working area*) como sendo a região de recência onde o *working set* temporário está confinado, isto é, a porção inicial da fila LRU em que todos os acessos realizados entre as duas últimas faltas de página se concentraram.

O algoritmo LRU-WAR inova ao monitorar os acessos à memória e verificar, entre duas faltas consecutivas, a página referenciada com menor recência em relação ao seu último acesso; ou seja, ele identifica a posição mais alta da fila LRU que recebeu acessos nesse período. Tal posição, representada por W, limita a área de trabalho. Logo, a área de trabalho consiste no trecho da fila LRU compreendido entre a última página referenciada – página MRU, início da fila – e a página que ocupa a W-ésima posição da fila. Uma conclusão importante é que o *working set* temporário contém sempre, no máximo, as páginas que ocupam esta área. A figura 1 exemplifica um caso hipotético de execução, no qual W é identificado, assim como a localização da área de trabalho.

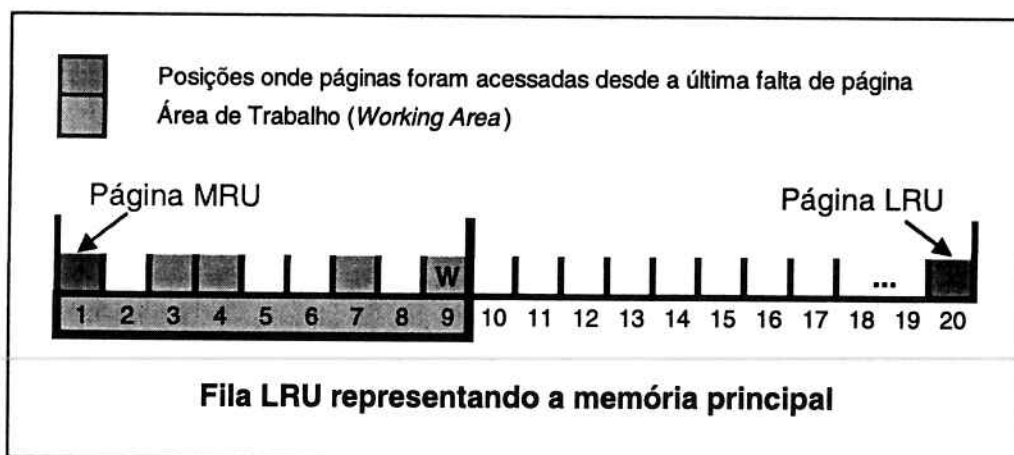


Fig. 1: Exemplo de área de trabalho, identificando-se o limite W entre faltas de página

É essencial salientar que as células da fila LRU preenchidas com a cor cinza, na figura 1, indicam as posições que receberam acessos desde a última falta de página até o instante vigente. Isto não quer dizer que as páginas referenciadas continuam nestas mesmas posições. Tampouco quer dizer que as páginas que atualmente as ocupam foram acessadas no período. De acordo com a política LRU, toda página referenciada é deslocada imediatamente para o início da fila. Os destaques em cinza especificam apenas um histórico das posições que tais páginas – não importa quais são – ocupavam no instante do seu respectivo acesso, antes da conseqüente reordenação na fila LRU e dentro do período de tempo considerado.

O algoritmo LRU-WAR trabalha basicamente com dois critérios de substituição: o critério LRU, no qual a página acessada há mais tempo é retirada da memória, e o critério MRU- n , no qual a n -ésima página mais recentemente acessada é descartada. O critério de substituição padrão é o LRU, que só deixa de ser utilizado quando uma tendência de padrão seqüencial de acessos se fortalece. O algoritmo detecta indícios de acesso seqüencial muito rapidamente, porém mantém o critério LRU e aguarda durante um certo período de carência até que a tendência se intensifique. O critério de substituição então passa a ser o MRU- n nesta situação. A página substituída por tal critério é sempre aquela que ocupa a primeira posição da fila LRU fora da área de trabalho. Assim, $n=W+1$; o que permite especificar a versão MRU- n utilizada como sendo MRU- $(W+1)$.

A idéia é substituir uma página que, por tendência, esteja se tornando inativa; ou seja, uma página que deixou de pertencer ao *working set* temporário do processo há pouco tempo. Com este ponto de substituição precoce, um grande número de páginas permanecem residentes na memória – todas aquelas que ocupam uma posição na fila LRU superior a $W+1$ – até que a tendência de acessos seqüenciais deixe de existir e o critério de substituição LRU volte a ser empregado.

3.2. Detalhes Operacionais

Para identificar tendências seqüenciais durante a execução de um programa, o algoritmo LRU-WAR verifica, a cada falta de página, se o tamanho da área de trabalho é pequeno o suficiente para sugerir que as páginas residentes estão sendo subutilizadas. Se for, assume a tendência seqüencial e monitora a variação de tamanho desta área durante as faltas seguintes. Permanecendo a área de trabalho pequena o suficiente por um certo número de faltas de página, entra em vigor o modo de operação seqüencial do algoritmo. Uma área de trabalho é considerada pequena quando está contida na região seqüencial, a qual corresponde às L (um dos dois parâmetro do algoritmo) primeiras posições da fila LRU. Consideramos o valor $L=\text{MIN}[50, M/2]$ bastante razoável e adequado ao bom funcionamento do LRU-WAR, sendo M o tamanho de memória disponível. Trata-se de um valor pequeno, mas não tão pequeno ao ponto de invalidar a contribuição do algoritmo. Este foi o valor adotado em nossos experimentos.

A região LRU complementa a região seqüencial e consiste, obviamente, na porção final da fila LRU. Uma subregião também é identificada dentro da região seqüencial: a chamada região protegida, compreendendo as C primeiras páginas da fila; C (carência mínima) é o outro parâmetro do algoritmo, cujo valor original é $C=5$. Esta pequena região engloba algumas posições de recência consideradas baixas demais para conter a área de trabalho, o que permite dizer que a menor área de trabalho aceita pelo algoritmo LRU-WAR é $C+1$. Logo, se $W \leq C$, o algoritmo atualiza o seu valor para $W=C+1$. A figura 2 ilustra as divisões lógicas na fila LRU que orientam o algoritmo.

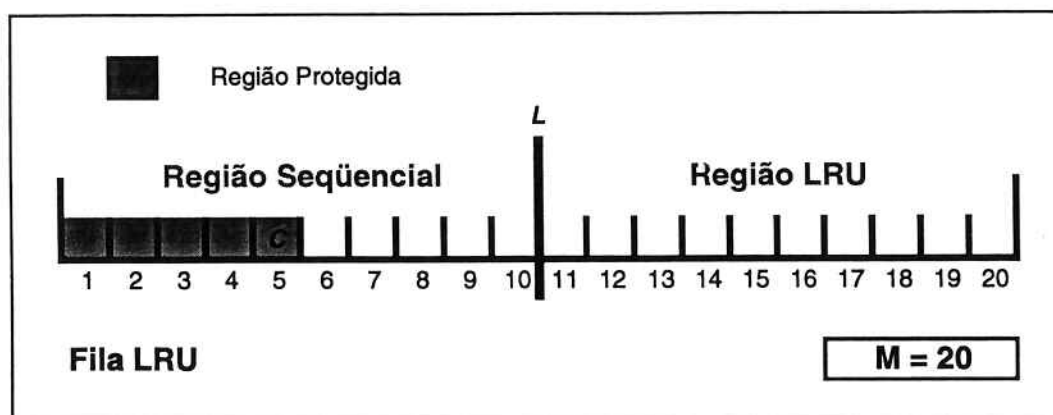


Fig. 2: Divisão lógica da fila LRU utilizada pelo algoritmo LRU-WAR

Uma tendência LRU é assumida quando, entre duas faltas de página consecutivas, o tamanho da área de trabalho ultrapassa o tamanho da região seqüencial. Nesta situação, o algoritmo zera a posição W – que representa o tamanho da área de trabalho – no momento da falta. Se no próximo instante em que uma falta de página ocorrer, a área de trabalho for maior que a região seqüencial novamente, a tendência continua sendo LRU e a posição W volta a ser zerada. Caso contrário, uma tendência seqüencial é detectada – a ser confirmada pela imutabilidade no tamanho da área de trabalho durante as faltas subseqüentes.

O algoritmo LRU-WAR entra em operação seqüencial somente após:

- A ocorrência de um número de faltas de página proporcional ao tamanho da área de trabalho – e, portanto, equivalente a W – desde o instante em que a tendência seqüencial começou;
- A ocorrência de um número adicional de faltas de página, após a primeira condição ter sido satisfeita, referente a um tempo de carência complementar, utilizado como meio de detectar e prevenir erros de decisão do algoritmo.

Isto, é claro, se a área de trabalho permanecer na região seqüencial durante todo este período, configurando uma baixa reutilização de páginas. Por exemplo, se a área de trabalho tem tamanho 10 em um determinado momento de execução ($W=10$), o critério MRU- n só é utilizado caso a área tenha se mantido menor ou igual a este tamanho durante as últimas 10 faltas de página, incluindo a atual, além do tempo de carência.

Fora de tendência LRU, ou seja, estando a área de trabalho contida na região seqüencial, a posição W somente aumenta, nunca diminui. Em outras palavras: em tendência ou operação seqüencial, a área de trabalho não é sensível à diminuição de tamanho da região de recência em que o *working set* temporário está confinado. Ela só é sensível a um aumento neste tamanho. Isto acontece porque o objetivo de W é limitar a região de recência em que todos os acessos estiveram concentrados desde que a tendência seqüencial foi detectada. A tabela I destaca os três estados de execução previstos pelo algoritmo e reitera os critérios de substituição empregados em cada caso.

Tabela I – Características dos estados de execução definidos pelo algoritmo LRU-WAR

ALGORITMO LRU-WAR			
Estado de execução	Tamanho da Área de Trabalho	Critério de Substituição	Ponto de Substituição (Posição na Fila LRU)
Tendência Original (LRU)	Maior que L	LRU	M
Tendência Seqüencial	Menor ou igual a L	LRU	M
Operação Seqüencial	Menor ou igual a L e estável	MRU- n	$W+1$

3.3. Descrição Prática do Algoritmo

O algoritmo LRU-WAR pode ser escrito na forma do seguinte pseudo-código:

```

1. Se a página acessada está carregada na memória:
2.   P = posição da página na fila LRU;
3.   Se P > W:
4.     Se N > 0: /* Se está em operação seqüencial */
5.       INÉRCIA = 0; /* Encerra a operação seqüencial */
6.       Se P > W+1 e P ≤ W+TC+1 e (N ≤ M-P ou N < 50):
7.         /* Detecta erro e aumenta o tempo de carência */
8.         TC = TC + N;
9.         N = 0;
10.      W = P; /* Aumenta área de trabalho */
11.      Reordena a página na primeira posição da fila.
12. Senão (a página acessada não está carregada na memória):
13.   Se a memória está cheia:
14.     Se W ≤ L:
15.       INÉRCIA = INÉRCIA + 1;
16.       Se W ≤ C:
17.         W = C + 1;
18.       Se INÉRCIA ≥ W+TC: /* Operação seqüencial */
19.         Se N < M ou N < 50:
20.           N = N + 1;
21.         Se TC > C:
22.           TC = TC - 1;
23.         Remove a página na posição W+1 da fila;
24.       Senão (INÉRCIA < W+TC): /* Tendência seqüencial */
25.         Remove a página na posição M da fila;
26.       Senão (W > L): /* Tendência LRU */
27.         INÉRCIA = 0;
28.         W = 0;
29.         N = 0;
30.         Remove a página na posição M da fila;
31. Carrega a página acessada no início da fila.

```

O tempo de carência TC consiste em um número de faltas de página adicional que o algoritmo respeita antes de operar em modo seqüencial. Apesar de a tendência seqüencial já ser bastante forte em dado momento, suficiente para o algoritmo começar a utilizar o critério de substituição MRU-n, o tempo de carência poupa algumas páginas que seriam substituídas de imediato por este critério, fazendo com que elas permaneçam residentes na memória.

O trecho compreendido entre as linhas de código 2 e 11 é executado quando uma página já residente na memória é acessada (*hit*). Neste caso, se a página ocupa uma posição P na fila LRU maior que a área de trabalho, o tamanho da área é aumentado, passando a englobar esta posição: $W = P$. Em operação seqüencial, a contagem de faltas de página é reiniciada, encerrando tal fase. O modo de operação volta a ser, então, tendência LRU ou tendência seqüencial, dependendo do tamanho da nova área de trabalho. Se um erro de decisão é detectado, ou seja, uma das páginas preservadas da substituição seqüencial pelo tempo de carência foi a responsável pelo aumento de W, e a sua reutilização foi considerada muito rápida, o tempo de carência TC aumenta de acordo com o número de substituições realizadas em operação seqüencial.

O trecho de código logo abaixo, entre as linhas 14 e 30, descreve a atuação do algoritmo quando uma substituição de página precisa ser realizada. Em tendência

original (LRU), a página acessada há mais tempo – que ocupa a posição M , última da fila – é substituída e a área de trabalho é zerada. Em tendência sequencial, o critério de substituição LRU também é empregado e a área de trabalho apenas pode aumentar de tamanho. Finalmente, se o algoritmo está operando em modo sequencial, a página que ocupa a posição $W+1$ da fila é substituída e a área de trabalho precisa se manter estável para que a operação não seja interrompida. Ressalte-se que o tamanho da área de trabalho deve sempre ultrapassar o tamanho da região protegida, condição garantida pelas linhas 16 e 17. O número de faltas de página, desde o começo de uma tendência sequencial, é calculado através do contador INÉRCIA. Por sua vez, o contador N é responsável pelo armazenamento do número de faltas de página computadas em operação sequencial.

Uma versão *online* (factível) do algoritmo LRU-WAR também foi criada, com a mesma simplicidade algorítmica da versão *offline* (completa, mas teórica). Assim como ocorre nas implementações reais do LRU, esta versão *online* do LRU-WAR é apenas uma aproximação da versão *offline*. A grande diferença está na fila LRU, que não pode ser atualizada em todos os acessos, apenas durante os momentos de falta de página.

Justificativas a respeito de cada decisão de projeto e explicações minuciosas sobre aspectos operacionais do algoritmo LRU-WAR, inclusive de sua versão *online*, podem ser encontradas com mais detalhes em Cassettari (2004).

4. AVALIAÇÃO DE DESEMPENHO

Um arquivo de *traces* consiste basicamente na listagem cronológica dos diversos acessos à memória realizados por um processo. Em outras palavras, descreve passo a passo o comportamento de um dado programa em termos de utilização da memória. Estes arquivos são obtidos através de um gerador de *traces*, que executa o programa alvo e coleta os acessos à memória realizados ao longo de seu processamento (UHLIG; MUDGE, 1997).

A partir dos dados coletados e armazenados nos arquivos de *traces*, uma série de experiências podem ser efetuadas para se avaliar o desempenho de algoritmos de substituição, no contexto de todos os possíveis tamanhos de memória que se queira considerar. Utilizando esta técnica de avaliação, um conjunto de simulações foi organizado para averiguar o desempenho do algoritmo LRU-WAR e compará-lo com a política de substituição de páginas LRU original. O critério de desempenho utilizado é o número de faltas de página geradas na execução de um processo.

4.1. Simulações Realizadas

Os experimentos aqui relatados tiveram como carga de simulação os 7 arquivos de *trace* que compõem o pacote VMTrace, originalmente aplicados por Smaragdakis; Kaplan; Wilson (1999) nas simulações do algoritmo EELRU, sendo também por eles disponibilizados. O nome deste pacote designa a ferramenta de geração de *traces* utilizada em sua captação, desenvolvida pelos próprios autores. A seguir são resumidamente descritos os programas representados por estes 7 arquivos:

- *Espresso*: Simulador de circuito;
- *GCC*: Compilador C/C++ do projeto GNU, versão 2.7.2;
- *Gnuplot*: Gerador de gráficos do projeto GNU;
- *Grobner*: Programa matemático que trabalha com Bases de Gröbner;

- *GS*: GhostScript 3.33, interpretador PostScript;
- *Lindsay*: Simulador de hipercubo;
- *P2C*: Tradutor de programas em Pascal para C.

O programa Gnuplot é o único que apresenta padrões sequenciais de acesso à memória bem definidos, intercalado com acessos a poucas páginas que exibem alta localidade temporal. Em contrapartida, os demais arquivos VMTrace se caracterizam principalmente pelo aspecto da localidade de referências, sem a presença de outros padrões regulares facilmente identificáveis.

Tabela II – Programas e contextos de memória considerados nas simulações realizadas

Simulações Realizadas					
Arquivo de Traces	Páginas Distintas	Tamanhos de Memória Simulados	Interv. entre Tamanhos	Número de Simulações	
VMTrace	Espresso	77	10, 11, 12, ..., 73, 74, 75	1	66
	GCC	458	10, 15, 20, ..., 445, 450, 455	5	90
	Gnuplot	7718	100, 200, 300, ..., 7500, 7600, 7700	100	77
	Grobner	67	10, 11, 12, ..., 63, 64, 65	1	56
	GS	558	10, 15, 20, ..., 545, 550, 555	5	110
	Lindsay	521	10, 15, 20, ..., 510, 515, 520	5	103
	P2C	132	10, 15, 20, ..., 120, 125, 130	5	25
	Total de Simulações				527

A tabela II descreve as baterias de simulação às quais os programas foram submetidos. A execução de cada programa é simulada em diferentes contextos de disponibilidade de memória, isto é, com diversos tamanhos de memória disponível. Tamanhos muito pequenos (abaixo de 10 páginas) e muito grandes (acima do número de páginas que o programa referencia em todo o seu processamento) foram desprezados, pois não permitem uma comparação efetiva entre os algoritmos – uma vez que os resultados de desempenho são equivalentes nestas condições.

A segunda coluna da tabela II informa o número de páginas distintas referenciadas pelos programas, enquanto a terceira coluna relata os tamanhos de memória utilizados em cada simulação e a quarta coluna indica o intervalo entre estes tamanhos. A última coluna, finalmente, contabiliza o número de simulações efetuadas. Cada simulação avalia o desempenho de um algoritmo considerando apenas um único tamanho de memória. O conjunto das simulações referentes ao programa indica o desempenho médio que o algoritmo obteve com o mesmo.

4.2. Resultados Obtidos

A tabela III demonstra percentuais de desempenho referentes à diferença entre o número de faltas de página gerado pelo algoritmo LRU e o número gerado pelo LRU-WAR. Valores percentuais positivos indicam que o novo algoritmo apresenta um desempenho pior que o do algoritmo LRU, pois provoca um número maior de faltas de página. Valores negativos, obviamente, indicam o contrário.

A coluna “melhor caso” representa a situação de execução em que o algoritmo LRU-WAR obtém o melhor desempenho percentual comparado com o algoritmo LRU. A situação inversa é representada pela coluna “pior caso”: quando o LRU obtém o melhor desempenho. A média aritmética das diferenças percentuais coletadas entre os dois algoritmos, levando-se em conta todas as simulações descritas na tabela II – todos os tamanhos de memória considerados –, é calculada na terceira coluna. Um exemplo:

56 simulações são realizadas para avaliar o desempenho médio de cada algoritmo na execução do programa Grobner e, portanto, 56 diferenças percentuais são calculadas entre números de faltas de página gerados pelo algoritmo LRU e números gerados pelo algoritmo LRU-WAR. A menor destas diferenças (maior redução no número de faltas de página) representa o melhor caso do algoritmo LRU-WAR em relação ao LRU original; a maior diferença representa o pior caso; e a média aritmética entre as 56 diferenças indica uma tendência geral.

Tabela III – Desempenho do algoritmo LRU-WAR em relação ao algoritmo LRU

ALGORITMO LRU-WAR: Diferença percentual no número de faltas de página em relação ao algoritmo LRU				
Arquivos de Traca	Melhor Caso		Pior Caso	
	Diferença % (Mem.)		Diferença % (Mem.)	
Espresso	-7,94%	(12)	0,02%	(27)
GCC	-9,66%	(185)	0,18%	(245)
Gnuplot	-66,22%	(7700)	-0,53%	(100)
Grobner	-11,75%	(28)	1,02%	(13)
GS	-5,61%	(155)	5,09%	(300)
Lindsay	-17,38%	(...)	1,74%	(130)
P2C	-0,72%	(50)	0,04%	(75)
Pacote VMTrace	-66,22%		5,09%	

Assim, podemos interpretar da seguinte forma os dados relativos ao programa Grobner: o algoritmo LRU-WAR, em média, atinge um desempenho 3,78% melhor que o do algoritmo LRU. Porém, ele pode provocar um número de faltas de página até 1,02% maior que o mesmo, o que acontece especificamente com uma memória de tamanho 13. Em uma outra situação de execução (memória disponível de 28 páginas), entretanto, ele exibe um resultado 11,75% melhor. A conclusão é que o LRU-WAR pode ser melhor ou pior que o LRU no processamento de tal programa, dependendo do tamanho de memória disponível; mas a média sugere que ele tende a apresentar melhores resultados, o que é confirmado pela figura 3.

Os gráficos da figura 3 informam a variação percentual de desempenho do algoritmo LRU-WAR em relação aos resultados apresentados pelo algoritmo LRU nas simulações efetuadas com dois programas de características distintas: Grobner e Gnuplot. O programa Grobner representa uma situação típica: o algoritmo LRU-WAR é melhor do que o algoritmo LRU, mas não muito melhor. Isto ocorre quando o processo apresenta trechos com acessos sequenciais, mas tal padrão não é predominante, o que dificulta a sua detecção por parte do LRU-WAR. Por outro lado, o programa Gnuplot exibe um padrão efetivo de acessos sequenciais à memória, que predomina do começo ao fim de seu processamento. Neste caso, o algoritmo LRU apresenta um desempenho muito ruim, enquanto o LRU-WAR gera excelentes resultados.

A conclusão a que se chega, a partir dos resultados práticos obtidos, é que o algoritmo LRU-WAR atende ao seu propósito: é significativamente melhor do que o algoritmo LRU quando padrões de acesso sequenciais vigoram e não é muito pior em nenhuma situação analisada. No seu melhor caso, o algoritmo LRU-WAR alcançou um desempenho 66,22% superior ao do LRU. Não obstante, seu pior desempenho em relação ao mesmo gerou uma diferença percentual máxima de 5,09% no número de faltas de página, em um universo de 527 simulações realizadas.

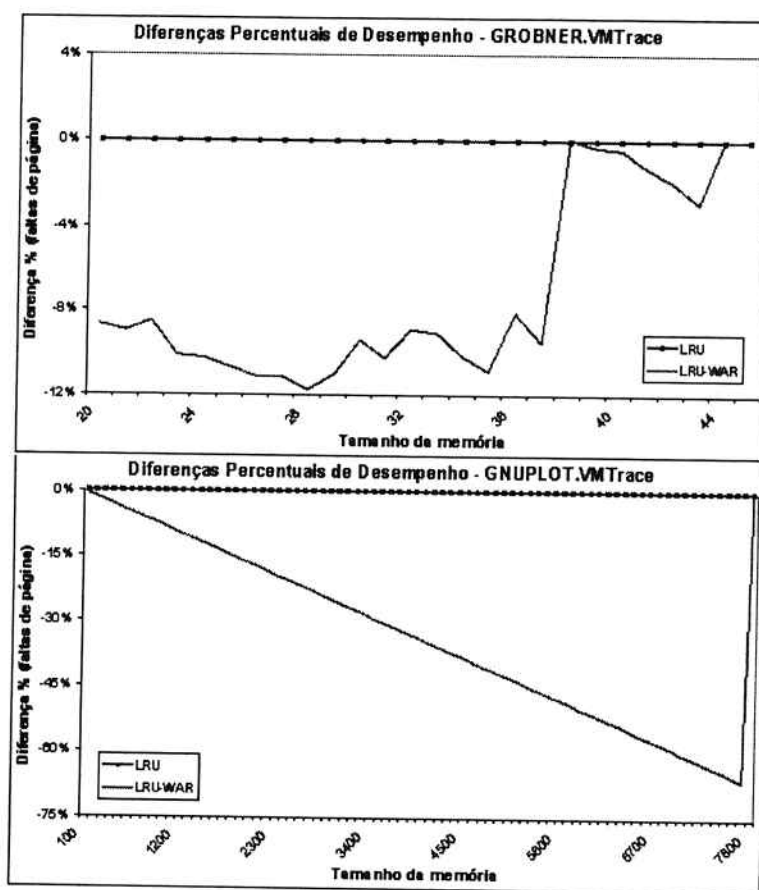


Fig. 3: Gráficos de desempenho do algoritmo LRU-WAR em relação ao algoritmo LRU

5. CONCLUSÃO

Após uma importante análise acerca das propriedades do modelo LRU e de algumas características inerentes a programas dos mais variados, verificamos que poderíamos reconhecer padrões de acesso sequenciais, dinamicamente, somente identificando as posições na fila LRU ocupadas pelas páginas referenciadas ao longo do tempo por um processo. A nova técnica poderia, assim, contribuir para melhorar o desempenho do algoritmo LRU original na presença de tais padrões, visto que acessos sequenciais tendem a romper a eficiência normalmente observada em sua atuação.

Vários aspectos operacionais foram então definidos, dando origem ao algoritmo adaptativo de substituição de páginas LRU-WAR (LRU com Confinamento da Área de Trabalho), uma proposta viável e inédita apresentada neste artigo. Simulações demonstraram que o algoritmo LRU-WAR alcança desempenhos excelentes quando trabalha com programas que apresentam padrões de acesso basicamente sequenciais. Ao mesmo tempo, o algoritmo se mostrou confiável, uma vez que os seus resultados não foram significativamente piores que os do LRU tradicional em nenhum caso analisado.

Um estudo de avaliação de desempenho muito mais completo e detalhado está disponível em Cassettari (2004), que também descreve com maior profundidade as motivações, parâmetros, estruturas e considerações operacionais que fundamentam e orientam o algoritmo LRU-WAR. A implementação prática da versão *online* do novo algoritmo em um sistema operacional de código aberto, provavelmente o Linux, é a seqüência natural deste trabalho.

LISTA DE REFERÊNCIAS

CASSETTARI, H.H. *Análise da localidade de programas e desenvolvimento de algoritmos adaptativos para substituição de páginas*. 2004. 118p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2004.

CHOI, J. et al. An implementation study of a detection-based adaptive block replacement scheme. In: ANNUAL TECHNICAL CONFERENCE, 4., Monterey, 1999. *USENIX' 99: Proceedings*. Monterey: USENIX, 1999. p.239-252.

_____. Towards application/file-level characterization of block references: a case for fine-grained buffer management. In: INTERNATIONAL CONFERENCE ON MEASUREMENT AND MODELING OF COMPUTER SYSTEMS, 25., Santa Clara, 2000. *SIGMETRICS' 00: Proceedings*. Santa Clara: ACM, 2000. p.286-295.

DENNING, P.J. The working set model for program behavior. *Communications of the ACM*, v.11, n.5, p.323-333, 1968.

_____. Working sets: past and present. *IEEE Transactions on Software Engineering*, v.SE-6, n.1, p.64-84, 1980.

GLASS, G.; CAO, P. Adaptive page replacement based on memory reference behavior. In: INTERNATIONAL CONFERENCE ON MEASUREMENT AND MODELING OF COMPUTER SYSTEMS, 22., Seattle, 1997. *SIGMETRICS' 97: Proceedings*. Seattle: ACM, 1997. p.115-126.

JIANG, S.; ZHANG, X. LIRS: An efficient low inter-reference recency set replacement policy to improve buffer cache performance. In: INTERNATIONAL CONFERENCE ON MEASUREMENT AND MODELING OF COMPUTER SYSTEMS, 27., Marina Del Rey, 2002. *SIGMETRICS' 02: Proceedings*. Marina Del Rey: ACM, 2002. p.31-42.

KIM, J.M. et al. A low-overhead high-performance unified buffer management scheme that exploits sequential and looping references. In: SYMPOSIUM ON OPERATING SYSTEM DESIGN AND IMPLEMENTATION, 4., San Diego, 2000. *OSDI' 2000: Proceedings*. San Diego: USENIX, 2000. p.119-134.

MIDORIKAWA, E.T. *Uma nova estratégia para a gerência de memória para sistemas de computação de alto desempenho*. 1997. 193p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 1997.

SMARAGDAKIS, Y.; KAPLAN, S.; WILSON, P. EELRU: simple and effective adaptive page replacement. In: INTERNATIONAL CONFERENCE ON MEASUREMENT AND MODELING OF COMPUTER SYSTEMS, 24., Atlanta, 1999. *SIGMETRICS' 99: Proceedings*. Atlanta: ACM, 1999. p.122-133.

UHLIG, R.A.; MUDGE, T.N. Trace-driven memory simulation: a survey. *ACM Computing Surveys*, v.29, n.2, p-128-170, 1997.

BOLETINS TÉCNICOS - TEXTOS PUBLICADOS

- BT/PCS/9301 - Interligação de Processadores através de Chaves Ômicron - GERALDO LINO DE CAMPOS, DEMI GETSCHKO
- BT/PCS/9302 - Implementação de Transparência em Sistema Distribuído - LUÍSA YUMIKO AKAO, JOÃO JOSÉ NETO
- BT/PCS/9303 - Desenvolvimento de Sistemas Especificados em SDL - SIDNEI H. TANO, SELMA S. S. MELNIKOFF
- BT/PCS/9304 - Um Modelo Formal para Sistemas Digitais à Nível de Transferência de Registradores - JOSÉ EDUARDO MOREIRA, WILSON VICENTE RUGGIERO
- BT/PCS/9305 - Uma Ferramenta para o Desenvolvimento de Protótipos de Programas Concorrentes - JORGE KINOSHITA, JOÃO JOSÉ NETO
- BT/PCS/9306 - Uma Ferramenta de Monitoração para um Núcleo de Resolução Distribuída de Problemas Orientado a Objetos - JAIME SIMÃO SICHMAN, ELERI CARDOSO
- BT/PCS/9307 - Uma Análise das Técnicas Reversíveis de Compressão de Dados - MÁRIO CESAR GOMES SEGURA, EDIT GRASSIANI LINO DE CAMPOS
- BT/PCS/9308 - Proposta de Rede Digital de Sistemas Integrados para Navio - CESAR DE ALVARENGA JACOBY, MOACYR MARTUCCI JR.
- BT/PCS/9309 - Sistemas UNIX para Tempo Real - PAULO CESAR CORIGLIANO, JOÃO JOSÉ NETO
- BT/PCS/9310 - Projeto de uma Unidade de Matching Store baseada em Memória Paginada para uma Máquina Fluxo de Dados Distribuído - EDUARDO MARQUES, CLAUDIO KIRNER
- BT/PCS/9401 - Implementação de Arquiteturas Abertas: Uma Aplicação na Automação da Manufatura - JORGE LUIS RISCO BECERRA, MOACYR MARTUCCI JR.
- BT/PCS/9402 - Modelamento Geométrico usando do Operadores Topológicos de Euler - GERALDO MACIEL DA FONSECA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9403 - Segmentação de Imagens aplicada a Reconhecimento Automático de Alvos - LEONCIO CLARO DE BARROS NETO, ANTONIO MARCOS DE AGUIRRA MASSOLA
- BT/PCS/9404 - Metodologia e Ambiente para Reutilização de Software Baseado em Composição - LEONARDO PUJATTI, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9405 - Desenvolvimento de uma Solução para a Supervisão e Integração de Células de Manufatura Discreta - JOSÉ BENEDITO DE ALMEIDA, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/9406 - Método de Teste de Sincronização para Programas em ADA - EDUARDO T. MATSUDA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/9407 - Um Compilador Paralelizante com Detecção de Paralelismo na Linguagem Intermediária - HSUEH TSUNG HSIANG, LÍRIA MATSUMOTO SAITO
- BT/PCS/9408 - Modelamento de Sistemas com Redes de Petri Interpretadas - CARLOS ALBERTO SANGIORGIO, WILSON V. RUGGIERO
- BT/PCS/9501 - Síntese de Voz com Qualidade - EVANDRO BACCI GOUVÊA, GERALDO LINO DE CAMPOS
- BT/PCS/9502 - Um Simulador de Arquiteturas de Computadores "A Computer Architecture Simulator" - CLAUDIO A. PRADO, WILSON V. RUGGIERO
- BT/PCS/9503 - Simulador para Avaliação da Confiabilidade de Sistemas Redundantes com Reparo - ANDRÉA LUCIA BRAGA, FRANCISCO JOSÉ DE OLIVEIRA DIAS
- BT/PCS/9504 - Projeto Conceitual e Projeto Básico do Nível de Coordenação de um Sistema Aberto de Automação, Utilizando Conceitos de Orientação a Objetos - NELSON TANOMARU, MOACYR MARTUCCI JUNIOR
- BT/PCS/9505 - Uma Experiência no Gerenciamento da Produção de Software - RICARDO LUIS DE AZEVEDO DA ROCHA, JOÃO JOSÉ NETO
- BT/PCS/9506 - MetodOO - Método de Desenvolvimento de Sistemas Orientado a Objetos: Uma Abordagem Integrada à Análise Estruturada e Redes de Petri - KECHI HIRAMA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/9601 - MOOPP: Uma Metodologia Orientada a Objetos para Desenvolvimento de Software para Processamento Paralelo - ELISA HATSUE MORIYA HUZITA, LÍRIA MATSUMOTO SATO
- BT/PCS/9602 - Estudo do Espalhamento Brillouin Estimulado em Fibras Ópticas Monomodo - LUIS MEREGE SANCHES, CHARLES ARTUR SANTOS DE OLIVEIRA
- BT/PCS/9603 - Programação Paralela com Variáveis Compartilhadas para Sistemas Distribuídos - LUCIANA BEZERRA ARANTES, LÍRIA MATSUMOTO SATO
- BT/PCS/9604 - Uma Metodologia de Projeto de Redes Locais - TEREZA CRISTINA MELO DE BRITO CARVALHO, WILSON VICENTE RUGGIERO

- BT/PCS/9605 - Desenvolvimento de Sistema para Conversão de Textos em Fonemas no Idioma Português - DIMAS TREVIZAN CHBANE, GERALDO LINO DE CAMPOS
- BT/PCS/9606 - Sincronização de Fluxos Multimídia em um Sistema de Videoconferência - EDUARDO S. C. TAKAHASHI, STEFANIA STIUBIENER
- BT/PCS/9607 - A importância da Completeza na Especificação de Sistemas de Segurança - JOÃO BATISTA CAMARGO JÚNIOR, BENÍCIO JOSÉ DE SOUZA
- BT/PCS/9608 - Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança - BRÁULIO COELHO ÁVILA, MÁRCIO RILLO
- BT/PCS/9609 - Implementação de Engenharia Simultânea - MARCIO MOREIRA DA SILVA, MOACYR MARTUCCI JÚNIOR
- BT/PCS/9610 - Statecharts Adaptativos - Um Exemplo de Aplicação do STAD - JORGE RADY DE ALMEIDA JUNIOR, JOÃO JOSÉ NETO
- BT/PCS/9611 - Um Meta-Editor Dirigido por Sintaxe - MARGARETE KEIKO IWAI, JOÃO JOSÉ NETO
- BT/PCS/9612 - Reutilização em Software Orientado a Objetos: Um Estudo Empírico para Analisar a Dificuldade de Localização e Entendimento de Classes - SELMA SHIN SHIMIZU MELNIKOFF, PEDRO ALEXANDRE DE OLIVEIRA GIOVANI
- BT/PCS/9613 - Representação de Estruturas de Conhecimento em Sistemas de Banco de Dados - JUDITH PAVÓN MENDONZA, EDIT GRASSIANI LINO DE CAMPOS
- BT/PCS/9701 - Uma Experiência na Construção de um Tradutor Inglês - Português - JORGE KINOSHITA, JOÃO JOSÉ NETO
- BT/PCS/9702 - Combinando Análise de "Wavelet" e Análise Entrópica para Avaliar os Fenômenos de Difusão e Correlação - RUI CHUO HUEI CHIOU, MARIA ALICE G. V. FERREIRA
- BT/PCS/9703 - Um Método para Desenvolvimento de Sistemas de Computacionais de Apoio a Projetos de Engenharia - JOSÉ EDUARDO ZINDEL DEBONI, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/9704 - O Sistema de Posicionamento Global (GPS) e suas Aplicações - SÉRGIO MIRANDA PAZ, CARLOS EDUARDO CUGNASCA
- BT/PCS/9705 - METAMBI-OO - Um Ambiente de Apoio ao Aprendizado da Técnica Orientada a Objetos - JOÃO UMBERTO FURQUIM DE SOUZA, SELMA S. S. MELNIKOFF
- BT/PCS/9706 - Um Ambiente Interativo para Visualização do Comportamento Dinâmico de Algoritmos - IZAURA CRISTINA ARAÚJO, JOÃO JOSÉ NETO
- BT/PCS/9707 - Metodologia Orientada a Objetos e sua Aplicação em Sistemas de CAD Baseado em "Features" - CARLOS CÉSAR TANAKA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9708 - Um Tutor Inteligente para Análise Orientada a Objetos - MARIA EMÍLIA GOMES SOBRAL, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/9709 - Metodologia para Seleção de Solução de Sistema de Aquisição de Dados para Aplicações de Pequeno Porte - MARCELO FINGUERMAN, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/9801 - Conexões Virtuais em Redes ATM e Escalabilidade de Sistemas de Transmissão de Dados sem Conexão - WAGNER LUIZ ZUCCHI, WILSON VICENTE RUGGIERO
- BT/PCS/9802 - Estudo Comparativo dos Sistemas da Qualidade - EDISON SPINA, MOACYR MARTUCCI JR.
- BT/PCS/9803 - The VIBRA Multi-Agent Architecture: Integrating Purposive Vision With Deliberative and Reactive Planning - REINALDO A. C. BIANCHI, ANNA H. REALI C. RILLO, LELIANE N. BARROS
- BT/PCS/9901 - Metodologia ODP para o Desenvolvimento de Sistemas Abertos de Automação - JORGE LUIS RISCO BECCERRA, MOACYR MARTUCCI JUNIOR
- BT/PCS/9902 - Especificação de Um Modelo de Dados Bitemporal Orientado a Objetos - SOLANGE NICE ALVES DE SOUZA, EDIT GRASSIANI LINO DE CAMPOS
- BT/PCS/9903 - Implementação Paralela Distribuída da Dissecção Cartesiana Aninhada - HILTON GARCIA FERNANDES, LIRIA MATSUMOTO SATO
- BT/PCS/9904 - Metodologia para Especificação e Implementação de Solução de Gerenciamento - SERGIO CLEMENTE, TEREZA CRISTINA MELO DE BRITO CARVALHO
- BT/PCS/9905 - Modelagem de Ferramenta Hipermídia Aberta para a Produção de Tutoriais Interativos - LEILA HYODO, ROMERO TORI
- BT/PCS/9906 - Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com Dois Valores-LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos - JOÃO I. DA SILVA FILHO, JAIR MINORO ABE
- BT/PCS/9907 - Modelo Nebuloso de Confiabilidade Baseado no Modelo de Markov - PAULO SÉRGIO CUGNASCA, MARCO TÚLIO CARVALHO DE ANDRADE

- BT/PCS/9908 – Uma Análise Comparativa do Fluxo de Mensagens entre os Modelos da Rede Contractual (RC) e Colisões Baseada em Dependências (CBD) – MÁRCIA ITO, JAIME SIMÃO SICHMAN
- BT/PCS/9909 – Otimização de Processo de Inserção Automática de Componentes Eletrônicos Empregando a Técnica de Times Assíncronos – CESAR SCARPINI RABAK, JAIME SIMÃO SICHMAN
- BT/PCS/9910 – MIISA – Uma Metodologia para Integração da Informação em Sistemas Abertos – HILDA CARVALHO DE OLIVEIRA, SELMA S. S. MELNIKOFF
- BT/PCS/9911 – Metodologia para Utilização de Componentes de Software: um estudo de Caso – KAZUTOSI TAKATA, SELMA S. S. MELNIKOFF
- BT/PCS/0001 – Método para Engenharia de Requisitos Norteados por Necessidades de Informação – ARISTIDES NOVELLI FILHO, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0002 – Um Método de Escolha Automática de Soluções Usando Tecnologia Adaptativa – RICARDO LUIS DE AZEVEDO DA ROCHA, JOÃO JOSÉ NETO
- BT/PCS/0101 – Gerenciamento Hierárquico de Falhas – JAMIL KALIL NAUFAL JR., JOÃO BATISTA CAMARGO JR.
- BT/PCS/0102 – Um Método para a Construção de Analisadores Morfológicos, Aplicado à Língua Portuguesa, Baseado em Autômatos Adaptativos – CARLOS EDUARDO DANTAS DE MENEZES, JOÃO JOSÉ NETO
- BT/PCS/0103 – Educação pela Web: Metodologia e Ferramenta de Elaboração de Cursos com Navegação Dinâmica – LUISA ALEYDA GARCIA GONZÁLEZ, WILSON VICENTE RUGGIERO
- BT/PCS/0104 – O Desenvolvimento de Sistemas Baseados em Componentes a Partir da Visão de Objetos – RENATA EVANGELISTA ROMARIZ RECCO, JOÃO BATISTA CAMARGO JÚNIOR
- BT/PCS/0105 – Introdução às Gramáticas Adaptativas – MARGARETE KEIKO IWAI, JOÃO JOSÉ NETO
- BT/PCS/0106 – Automação dos Processos de Controle de Qualidade da Água e Esgoto em Laboratório de Controle Sanitário – JOSÉ BENEDITO DE ALMEIDA, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/0107 – Um Mecanismo para Distribuição Segura de Vídeo MPEG – CÍNTIA BORGES MARGI, GRAÇA BESSAN, WILSON VICENTE RUGGIERO
- BT/PCS/0108 – A Dependence-Based Model for Social Reasoning in Multi-Agent Systems – JAIME SIMÃO SICHMAN
- BT/PCS/0109 – Ambiente Multilinguagem de Programação – Aspectos do Projeto e Implementação – APARECIDO VALDEMIR DE FREITAS, JOÃO JOSÉ NETO
- BT/PCS/0110 – LETAC: Técnica para Análise de Tarefas e Especificação de Fluxo de Trabalho Cooperativo – MARCOS ROBERTO GREINER, LUCIA VILELA LEITE FILGUEIRAS
- BT/PCS/0111 – Modelagem ODP para o Planejamento de Sistemas de Potência – ANIRIO SALLES FILHO, JOSÉ SIDNEI COLOMBO MARTINI
- BT/PCS/0112 – Técnica para Ajuste dos Coeficientes de Quantização do Padrão MPEG em Tempo Real – REGINA M. SILVEIRA, WILSON V. RUGGIERO
- BT/PCS/0113 – Segmentação de Imagens por Classificação de Cores: Uma Abordagem Neural – ALEXANDRE S. SIMÕES, ANNA REALI COSTA
- BT/PCS/0114 – Uma Avaliação do Sistema DSM Nautilus – MARIO DONATO MARINO, GERALDO LINO DE CAMPOS
- BT/PCS/0115 – Utilização de Redes Neurais Artificiais para Construção de Imagem em Câmara de Cintilação – LUIZ SÉRGIO DE SOUZA, EDITH RANZINI
- BT/PCS/0116 – Simulação de Redes ATM – HSU CHIH WANG CHANG, WILSON VICENTE RUGGIERO
- BT/PCS/0117 – Application of Monoprocessed Architecture for Safety Critical Control Systems – JOSÉ ANTONIO FONSECA, JORGE RADY DE ALMEIDA JR.
- BT/PCS/0118 – WebBee – Um Sistema de Informação via WEB para Pesquisa de Abelhas sem Ferrão – RENATO SOUSA DA CUNHA, ANTONIO MOURA SARAIVA
- BT/PCS/0119 – Parallel Processing Applied to Robot Manipulator Trajectory Planning – DENIS HAMILTON NOMIYAMA, LÍRIA MATSUMOTO SATO, ANDRÉ RIYUITI HIRAKAWA
- BT/PCS/0120 – Utilização de Padrão de Arquitetura de Software para a Fase de Projeto Orientado a Objetos – CRISTINA MARIA FERREIRA DA SILVA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/0121 – Agilizando Aprendizagem por Reforço Através do uso de Conhecimento sobre o Domínio – RENÉ PEGORARO, ANNA H. REALI COSTA
- BT/PCS/0122 – Modelo de Segurança da Linguagem Java Problemas e Soluções – CLAUDIO MASSANORI MATAYOSHI, WILSON VICENTE RUGGIERO
- BT/PCS/0123 – Proposta de um Agente CNM para o Gerenciamento Web de um Backbone ATM – FERNANDO FROTA REDÍGOLO, TEREZA CRISTINA MELO DE BRITO CARVALHO
- BT/PCS/0124 – Um Método de Teste de software Baseado em Casos Teste – SÉRGIO RICARDO ROTTA, KECHI HIRAMA

- BT/PCS/0201 – A Teoria Nebulosa Aplicada a uma Bicicleta Ergométrica para Fisioterapia – MARCO ANTONIO GARMS, MARCO TÚLIO CARVALHO DE ANDRADE
- BT/PCS/0202 – Synchronization Constraints in a Concurrent Object Oriented Programming Model – LAÍS DO NASCIMENTO SALVADOR, LIRIA MATSUMOTO SATO
- BT/PCS/0203 – Construção de um Ambiente de Dados sobre um Sistema de Arquivos Paralelos – JOSÉ CRAVEIRO DA COSTA NETO, LIRIA MATSUMOTO SATO
- BT/PCS/0204 – Maestro: Um Middleware para Suporte a Aplicações Distribuídas Baseadas em Componentes de Software – CLÁUDIO LUÍS PEREIRA FERREIRA, JORGE LUÍS RISCO BECERRA
- BT/PCS/0205 – Sistemas de Automação dos Transportes (ITS) Descritos Através das Técnicas de Modelagem RM-OPD (ITU-T) e UML (OMG) – CLÁUDIO LUIZ MARTE, JORGE LUÍS RISCO BECERRA, JOSÉ SIDNEI COLOMBO
- BT/PCS/0206 – Comparação de Perfis de Usuários Coletados Através do Agente de Interface PersonalSearcher – GUSTAVO A. GIMÉNEZ LUGO, ANALÍA AMANDI, JAIME SIMÃO SICHMAN
- BT/PCS/0207 – Arquitetura Reutilizáveis para a Criação de Sistemas de Tutorização Inteligentes – MARCO ANTONIO FURLAN DE SOUZA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0208 – Análise e Predição de Desempenho de Programas Paralelos em Redes de Estações de Trabalho – LIN KUAN CHING, LIRIA MATSUMOTO SATO
- BT/PCS/0209 – Previsões Financeiras Através de Sistemas Neuronebulosos – DANIEL DE SOUZA GOMES, MARCO TÚLIO CARVALHO DE ANDRADE
- BT/PCS/0210 – Proposta de Arquitetura Aberta de Central de Atendimento – ANA PAULA GONÇALVES SERRA, MOACYR MARTUCCI JÚNIOR
- BT/PCS/0211 – Alternativas de Implementação de Sistemas Nebulosos em Hardware – MARCOS ALVES PREDEBON, MARCO TÚLIO CARVALHO DE ANDRADE
- BT/PCS/0212 – Registro de Imagens de Documentos Antigos – VALGUIMA VICTORIA VIANA ODAKURA MARTINEZ, GERALDO LINO DE CAMPOS
- BT/PCS/0213 – Um Modelo de Dados Multidimensional – PEDRO WILLEMSSENS, JORGE RADY DE ALMEIDA JUNIOR
- BT/PCS/0214 – Autômatos Adaptativos no Tratamento Sintático de Linguagem Natural – CÉLIA YUMI OKANO TANIWAKI, JOÃO JOSÉ NETO
- BT/PCS/0215 – Fatores e Subfatores para Avaliação da Segurança em Software de Sistemas Críticos – JOÃO EDUARDO PROENÇA PÁSCOA, JOÃO BATISTA CAMARGO JÚNIOR
- BT/PCS/0216 – Derivando um Modelo de Projeto a Partir de um Modelo de Análise, com Base em Design Patterns J2EE – SERGIO MARTINS FERNANDES, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/0217 – Domínios Virtuais para Redes Móveis Ad Hoc: Um Mecanismo de Segurança – LEONARDO AUGUSTO MARTUCCI, TEREZA CRISTINA DE MELO BRITO CARVALHO
- BT/PCS/0218 – Uma Ferramenta para a Formulação de Consultas Baseadas em Entidades e Papéis – ANDRÉ ROBERTO DORETO SANTOS, EDIT GRASSIANI LINO CAMPOS
- BT/PCS/0219 – Avaliação de Performance de Arquiteturas para Computação de Alto Desempenho – KARIN STRAUSS, WILSON VICENTE RUGGIERO
- BT/PCS/0220 – BGLsim: Simulador de Sistema Completo para o Blue Gene/L – LUÍS HENRIQUE DE BARROS CEZE, WILSON VICENTE RUGGIERO
- BT/PCS/0221 – μ P: Uma Solução de Micropagamentos – PEDRO ANCONA LOPEZ MINDLIN, TEREZA CRISTINA MELO DE BRITO CARVALHO
- BT/PCS/0222 – Modelamento de Roteadores IP para Análise de Atraso – MARCELO BLANES, GRAÇA BRESSAN
- BT/PCS/0223 – Uma Biblioteca de Classes Utilizando Java 3D para o Desenvolvimento de Ambientes Virtuais Multi-Usuários - RICARDO NAKAMURA, ROMERO TORI
- BT/PCS/0224 – Interactive 3D Physics Experiments Through the Internet – ALEXANDRE CARDOSO, ROMERO TORI
- BT/PCS/0225 – Avaliação do Desempenho de Aplicações Distribuídas sob Duas Velocidades de Rede – AMILCAR ROSA PEREIRA, GERALDO LINO DE CAMPOS
- BT/PCS/0226 – Acompanhamento do Aprendizado do Aluno em Cursos a Distância através da WEB: Metodologias e Ferramentas – LUCIANA APARECIDA MARTINEZ ZAINA, GRAÇA BRESSAN
- BT/PCS/0227 – Um Ambiente Colaborativo para Simulação de Redes de Computadores - OSCAR DANTAS VILCACHAGUA, GRAÇA BRESSAN
- BT/PCS/0301 – Diretrizes para o Projeto de Base de Dados Distribuídas – PEDRO LUIZ PIZZIGATTI CORRÊA, JORGE RADY DE ALMEIDA JR.
- BT/PCS/0302 – Análise e Predição de Desempenho de Programas MPI em Redes de Estações de Trabalho – JEAN MARCOS LAINE, EDSON T. MIDORIKAWA

- BT/PCS/0303 – Padrões de Software para Tutores Inteligentes Cooperativos em Engenharia de Requisitos – MARIA EMILIA GOMES SOBRAL, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0304 – Performance Analysis and Prediction of Some MPI Communication Primitives – HÉLIO MARCI DE OLIVEIRA, EDSON TOSHIMI MIDORIKAWA
- BT/PCS/0305 – RM-ODP para Expressar o Licenciamento Nuclear – EDILSON DE ANDRADE BARBOSA, MOACYR MARTUCCI JUNIOR
- BT/PCS/0306 – Modelo de Avaliação para Métricas de Software – VINICIUS DA SILVA ALMENDRA, KECHI HIRAMA
- BT/PCS/0307 – Análise de Confiabilidade de Sistemas Redundantes de Armazenamento em Discos Magnéticos – ENDERSON FERREIRA, JORGE RADY DE ALMEIDA JUNIOR
- BT/PCS/0308 – Utilizando Realidade Virtual e Objetos Distribuídos na Construção de uma Ferramenta de Aprendizagem Colaborativa – O Projeto Piaget – ISMAR FRANGO SILVEIRA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0309 – Construção de Base de Conhecimento em Prolog a partir de Páginas HTML – WAGNER TOSCANO, EDSON SATOSHI GOMI
- BT/PCS/0310 – Verificação de Segurança em Confluência de Trajetórias de Aeronaves Utilizando Autômatos Híbridos – ÍTALO ROMANI DE OLIVEIRA, PAULO SÉRGIO CUGNASCA
- BT/PCS/0311 – Sistemas de Reconhecimento Biométrico Aplicados à Segurança de Sistemas de Informação – VILMAR DE SOUZA MACHADO, JORGE RADY DE ALMEIDA JUNIOR
- BT/PCS/0312 – Análise Comparativa de Arquiteturas Híbridas Intserv-Diffserv Utilizadas para Obtenção de QoS Fim-a-Fim em Redes IP – CARLOS A. A. BENITES, GRAÇA BRESSAN
- BT/PCS/0313 – Proposta para Otimização de Desempenho do Protocolo TCP em Redes Wireless 802.11 – ANDRÉ AGUIAR SANTANA, TEREZA CRISTINA DE MELO BRITO CARVALHO
- BT/PCS/0314 – Using the Moise + Model for a Cooperative Framework of MAS Reorganization – JOMI FRED HUBENER, JAIME SIMÃO SICHMAN
- BT/PCS/0315 – Ferramenta para Acompanhamento da Participação do Aluno em Sessões de Fórum Aplicada no Ensino a Distância via Web – GUSTAVO BIANCHI CINELLI, GRAÇA BRESSAN
- BT/PCS/0316 – Uma Infra-Estrutura para Agentes Arrematantes em Múltiplos Leilões Simultâneos – PAULO ANDRÉ LIMA DE CASTRO, JAIME SIMÃO SICHMAN
- BT/PCS/0317 – Reutilização de Software Através de Geração de Código e de Desenvolvimento de Componentes – Estudo de Caso – FÁBIO FÚRIA SILVA, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0318 – Detecção Automática das Transições de Corte e Fades – IZAURA CRISTINA ARAÚJO, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0319 – Educação a Distância e a Web Semântica: Modelagem Ontológica de Materiais e Objetos de Aprendizagem para Plataforma CoL – MOYSÉS DE ARAÚJO, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0320 – Análise da Aplicação dos Padrões TMN no Gerenciamento de Sistemas de CRM – SANDRO ANTÔNIO VICENTE, MOACYR MARTUCCI JR
- BT/PCS/0321 – Alinhamento de Corpus Bilingües: Modelos e Aplicações – JOSÉ FONTEBASSO, JORGE KINOSHITA
- BT/PCS/0322 – Arquitetura de Integração dos Sistemas CORBA e Fieldbus: Aplicação do Padrão ODP – DANTE LINCOLN CAROAJULCA TANTALEÁN, JORGE LUIS RISCO BECERRA
- BT/PCS/0401 – Resultados Obtidos com a Implantação de um Ambiente para o Desenvolvimento de uma Maturidade em Engenharia de Software – LUIZ RICARDO BEGOSSO, LUCIA VILELA FILGUEIRAS
- BT/PCS/0402 – Procedimentos para Elaboração do Modelo de Análise UML com Características de Testabilidade – ROGÉRIA CRISTIANE GRATÃO DE SOUZA, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/0403 – INTEREXPO3D – Uma Ferramenta para Geração de Exposições Virtuais 3D Interativas – ANDRÉA ZOTOVICI, ROMERO TORI
- BT/PCS/0404 – Metodologia para Especificação e Implementação de Solução de Gerenciamento – SÉRGIO CLEMENTI, TEREZA CRISTINA MELO DE BRITO CARVALHO
- BT/PCS/0405 – CPAR - Cluster: Um Ambiente de Execução para Clusters de Nós Mono e Multiprocessadores – GISELE DA SILVA CRAVEIRO, LIRIA MATSUMOTO SATO
- BT/PCS/0406 – Monitores de Execução de Software para Sistemas Similares de Mesma Funcionalidade – SÉRGIO RICARDO ROTA, JORGE RADY DE ALMEIDA JR.
- BT/PCS/0407 – Avaliação do Perigo de Colisão entre Aeronaves em Operação de Aproximação em Pistas de Aterrissagem Paralelas – PAULO HIDESHI OGATA, JOÃO BATISTA CAMARGO JR.
- BT/PCS/0408 – Integration of Ontologies and Organization Models: A MAS View – GUSTAVO GIMENEZ LUGO, JAIME SIMÃO SICHMAN
- BT/PCS/0409 – The use Heuristics to Speedup Reinforcement Learning – REINALDO AUGUSTO DA COSTA BIANCHI, ANNA HELENA REALI COSTA

- BT/PCS/0410 – Gestão de QOS na Visão ODP: Uma Aplicação na Arquitetura SLM – CRISTINA MORI MIYATA, JORGE LUIS RISCO BECERRA
- BT/PCS/0411 – Proposta de um Modelo Simplificado de Aquisição de Software para Pequenas Empresas – PAULO SÉRGIO BRANDÃO LIMA, LÚCIA VILELA LEITE FILGUEIRAS
- BT/PCS/0412 – Avaliação da Carga Mental de Trabalho na Operação de Interfaces Homem - Computador de Sistemas de Controle de Processo – JOSÉ LUIZ LOPES ALVES, LÚCIA VILELA LEITE FILGUEIRAS
- BT/PCS/0413 – Identificação de Aspectos de Desenho de Interface de Documentos Hipermedia Educacionais que Influenciam na aprendizagem e Propostas de Utilização – RENATO JOSUÉ DE CARVALHO, MARIA ALICE GRIGAS VARELLA FERREIRA
- BT/PCS/0501 – Representação Gramatical Adaptativa com Verificação de Aparência de Linguagens Dependentes de Contexto – CÉSAR ALBERTO BRAVO PARIENTE, JOÃO JOSÉ NETO
- BT/PCS/0502 – PartNet+: Simulando Parcerias entre Múltiplos Agentes – JULIO DE LIMA DO RÊGO MONTEIRO, JAIME SIMÃO SICHMAN
- BT/PCS/0503 – Um Processo de Transformação de Arquiteturas de Sistemas Legados Baseado em Reengenharia – RODRIGO ALVARES DE SOUZA, REGINALDO ARAKAKI
- BT/PCS/0504 – Uma Proposta de Ontologia para Plano de Projeto – LUIS ALVES FERREIRA FILHO, EDSON SATOSHI GOMI
- BT/PCS/0505 – Ensino de Arquitetura de Computadores Utilizando Simuladores Completos – ANDRÉ EVANDRO LOURENÇO, EDSON TOSHIMI MIDORIKAWA
- BT/PCS/0506 – Armazenamento de Documentos XML "Text-Centric" e "Data-Centric" em Sistemas Relacionais e Objeto-Relacionais – CLÓVIS KIYOHIDE HANASHIRO, EDIT GRASSIANI LINO DE CAMPOS
- BT/PCS/0507 – Integração ODP – CIM para Definição de Arquitetura de Sistema de Informação: Uma Aplicação na Qualidade de Distribuição de Energia Elétrica – PEDRO SOUZA ROSA, JORGE LUIS RISCO BECERRA
- BT/PCS/0508 – Uma Ferramenta de Apoio para a Gestão do Conhecimento com Base em Representação por Mapas – SIDNEI NICOLI, EDISON SPINA
- BT/PCS/0509 – Modelo de Serviços e Arquitetura para Alocação Dinâmica de Recursos com MPLS-TE – Vivian Bastos Dias, Tereza Cristina Melo de Brito Carvalho
- BT/PCS/0510 – Atena: Um Sistema para Suporte ao Planejamento na Área de Gestão de Projetos – FABRÍCIO JAILSON BARTH, EDSON SATOSHI GOMI
- BT/PCS/0511 – A Floor Controlled Centralized Voice Conference System for Distance Learning – GEORGE MARCEL M. A. SMETANA, TEREZA CRISTINA MELO DE BRITO CARVALHO
- BT/PCS/0512 – Avaliação da Representação do Componente de Software na FARCSOft – Ferramenta de Apoio à Reutilização de Componentes de Software – ANA CLAUDIA ROSSI, SELMA SHIN SHIMIZU MELNIKOFF
- BT/PCS/0513 – AdapTools: Aspectos de Implementação e Utilização – HERMES PISTORI, JOÃO JOSÉ NETO