UFPR – Bacharelado em Ciência da Computação
CI212 – Organização e Arquitetura de Computadores
Trabalho Semestral: Projeto de Hierarquia de Memória — 2016-2 17 de outubro de 2016

Você foi encarregado de projetar o circuito de dados da hierarquia de memória para um sistema embarcado que executa um conjunto limitado de aplicações. Para tanto, os engenheiros de aplicação lhe entregaram um conjunto de *traces* com as referências a dados dos programas. Cada um dos *traces* contém a sequência de endereços referenciados pelo programa, e um indicador para o tipo de referência, se leitura ou escrita.

Você deve usar o simulador de caches `dinero` para medir a taxa de faltas de cada uma das configurações de cache e com ele calcular o tempo médio de acesso. Ainda, após avaliar o tempo médio de acesso, a área ocupada pela cache, e a energia dissipada em cada configuração, você deve recomendar o *melhor projeto* aos engenheiros de produto.

Os engenheiros de projeto lhe fornecem um conjunto de estimativas precisas de área e consumo de energia, para a tecnologia disponível.

O espírito deste trabalho é fazer o que se conhece por *exploração do espaço de projeto.*

Para tanto, você recebe um conjunto de parâmetros de projeto que inclui a capacidade C da cache, o tamanho de bloco B, a associatividade A, e a política de escrita W. A primeira parte de sua tarefa é escolher algumas combinações de valores para esses parâmetros, de forma a minimizar o tempo médio de acesso à memória $\tau = g(C, B, A, W)$.

Os "projetos preferenciais" $\mathcal{P}$, obtidos das simulações com `dinero`, devem ser os mínima da superfície dada pela função $\tau = g(C, B, A, W)$:

$$\mathcal{P} = \min(\partial g/\partial C,\ \partial g/\partial B,\ \partial g/\partial A,\ \partial g/\partial W)$$

A lição a extrair das simulações é: *o que se aprende com os resultados da simulação? O que eles nos dizem? O que eles <u>não</u> nos dizem?*

Além dos parâmetros de projeto da cache, restrições quanto a área α, energia ϵ e tempo médio de acesso à memória τ devem ser respeitadas.

Isso posto, nossa definição de *melhor* é um projeto que minimiza a função $f(\alpha, \epsilon, \tau)$, enquanto respeita as restrições de projeto.

As curvas para tempo de acesso, energia e área, que são mostradas em 2D a partir da página 5, são na verdade superfícies, se "puxarmos as curvas para fora do papel".

A exploração do espaço de projeto, para encontrar o conjunto de melhores projetos $\mathcal{M}$ consiste em combinar todas as três superfícies (área, energia e tempo de acesso) e então encontrar as derivadas parciais nas três direções

$$\mathcal{M} = \min(\partial f/\partial \alpha,\ \partial f/\partial \epsilon,\ \partial f/\partial \tau)$$

e finalmente encontrar os mínima locais, ou o minimo global, se existir. No nosso caso, a combinação das funções é uma soma ponderada dos valores de (α, ϵ, τ).

Note que as noções de derivadas parciais e minimização estão sendo usadas aqui num sentido aproximado, especialmente porque os parâmetros são discretos (C,B,A são potências de dois), e não temos as funções contínuas que relacionam os demais parâmetros. Contudo, a exploração do espaço de projeto é análoga à busca dos mínima locais, ou mínimo global, numa superfície não-trivial.

Estes textos podem ajudar na compreensão do que está sendo solicitado:
`http://www.inf.ufpr.br/roberto/wscad09pcvc.pdf`, e
`http://www.inf.ufpr.br/pos/techreport/RT_DINF002_2011.pdf`.

Etapa 1 Copie os fontes da página da disciplina e os desempacote num diretório apropriado:

```
SEM=16seg
wget http://www.inf.ufpr.br/roberto/ci212/trab${SEM}.tgz
wget http://www.inf.ufpr.br/roberto/ci212/trab${SEM}_TRC.tar
tar xzvf trab${SEM}.tgz
tar xvf  trab${SEM}_TRC.tar
```

Se não existem, crie os diretórios `$HOME/bin` e `$HOME/man/man1`, compile e instale os executáveis e páginas de manual:

```
mkdir ~/bin
mkdir -p ~/man/man1
cd trab162
(cd dinero ; make ; make install)
(cd sched  ; make ; make install)
cp man/*.1 ~/man/man1
```

Se ocorrer algum erro de compilação, por favor avise ao professor.

Edite seu `~/.bashrc` e acrescente os novos caminhos às suas variáveis de ambiente, conforme abaixo:

```
export PATH=$PATH:$HOME/bin
export MANPATH=$MANPATH:$HOME/man
```

Etapa 2 Uma vez que o ambiente de simulação esteja instalado, defina os parâmetros de projeto da cache que devem ser investigados e planeje os experimentos. Veja a página de manual de `dinero` para os parâmetros de projeto da cache que podem ser determinados nas simulações.

Veja os *scripts* `dinero/run`, `dinero/LOOP` e `dinero/GREP` para ter uma ideia das possibilidades. Estudamos estes *scripts* no quarto laboratório de Software Básico, em 2016-1.

Cada simulação gera um ponto do espaço de projeto. São necessárias N simulações para verificar N valores de um determinado parâmetro. Cada simulação gera um arquivo com os resultados da simulação e os valores de interesse (taxa de faltas) devem ser extraídos do arquivo e tabulados.

A taxa de faltas obtida de uma simulação depende do *trace* usado naquela simulação porque cada *trace* corresponde a um programa distinto. Os *traces* estão no diretório `TRC`.

Com os valores tabulados, um gráfico com os resultados pode ser gerado para acelerar a análise e observar tendências (taxa de faltas $\times$ variação de um parâmetro). O *script* `plot.gp` recebe um conjunto de dados e produz um gráfico com o programa `gnuplot`. Veja o enorme conjunto de possibilidades com `man gnuplot`.

Uma vez que um conjunto pequeno de projetos foi escolhido (2–4 projetos), os experimentos devem ser repetidos com estes projetos para todos os *traces*, para garantir que sua escolha é a melhor para todos os aplicativos. Veja, no livro texto, como computar um número para o desempenho do seu projeto que realmente quantifique sua qualidade.

Etapa 3 Uma vez escolhidos poucos projetos para a cache (2–4 projetos), você deve verificar o desempenho do seu projeto com uma carga relativamente realística. Para tanto use o programa `sched` para escalonar a execução dos vários aplicativos na cache e verificar, sob esta carga, qual é o melhor projeto.

O programa `sched` 'costura' e escalona a execução de vários *traces* como se fosse o escalonador de um sistema operacional. O *quantum* Q define o intervalo em que cada processo executa. A cada Q ciclos, ocorre uma troca de contexto, e as referências de outro *trace* passam a

ser enviadas ao simulador. Q é o número de referências entre cada troca de contexto, sendo portanto uma aproximação do número de ciclos entre cada troca de contexto – se as instruções de acesso à memória são 25% do total, o intervalo entre as trocas de contexto é $4Q$ ciclos.

O diagrama abaixo mostra o escalonamento de três processos: o processo 1 executa as referências i até q e ocorre um re-escalonamento; o processo 2 executa as referências r até h e ocorre um re-escalonamento; o processo 3 executa as referências a até z e ocorre um re-escalonamento, quando então o processo 1 volta a executar na referência seguinte (r_1) àquela da sua última (q_1) execução.

$$i_1 j_1 k_1 \cdots p_1 q_1 \mid r_2 s_2 t_2 \cdots f_2 g_2 h_2 \mid a_3 b_3 c_3 \cdots x_3 y_3 z_3 \mid r_1 s_1 t_1 \cdots f_1 g_1 \mid \cdots$$

Execute `sched/run` para entender o escalonamento da 'execução' dos processos. Escolha um intervalo razoável para a troca de processos executando no processador. Este intervalo deve ser longo o bastante para amortizar o custo das trocas de contexto – no MIPS, no mínimo 34 registradores devem ser salvados do processo que sai, e outros 34 recuperados para o processo que entra.

Nesta etapa você deve simular uma carga com os vários processos executando no processador e escolher qual o melhor projeto da cache para a(s) carga(s) simuladas. Você pode variar o *quantum* – opções `-q` e `-v` – e a mistura de processos, até o limite de 8 processos concorrentes.

Se o processo que tem a pior taxa de faltas executar mais de uma vez no *schedule*, os resultados tenderão para uma cache com maior capacidade; se o processo com a menor taxa de faltas for executado mais de uma vez, os resultados tenderão para caches menores. Escolha com cuidado e justifique suas escolhas.

Etapa 4 Com base nos dados sobre área e energia, na página 5, reveja a qualidade das suas escolhas na proposta de projeto de cache. Por exemplo, seu projeto pode atingir uma taxa de acertos excelente, a um custo inaceitável em termos de área e/ou energia.

O tempo de acesso do segundo nível é 60ns. Custa 1.0ns para transferir cada palavra do nível 2 para o nível 1.

O tempo de acesso da L1 (t) deve ser depreciado com a expressão

$$q = 0.25t + t^2$$

e q deve ser usado para computar o tempo médio de acesso à memória (τ). Qual é o tempo de acesso favorecido pela depreciação? O que os engenheiros de projeto estão tentando lhe dizer? O que os dados da especificação dizem?

Na avaliação das alternativas de projeto, os parâmetros têm os seguintes pesos:
área α: 40%
energia ϵ: 40%
tempo médio de acesso τ: 20%
Seu projeto deve minimizar, simultaneamente, o tempo médio de acesso, a área, e a energia.

O diretório `plots` contém um *script* (`blocs.gp`) e um conjunto de dados (`blocs.dat`) para plotar um gráfico de "taxa de faltas X tamanho dos blocos" usando o programa `gnuplot`.

Você deverá adaptar o *script* para mostrar seus resultados. Se você preferir gerar os gráficos com uma planilha, isso também é aceitável.

Especificação

1. O trabalho pode ser efetuado em duplas;
2. o relatório `xx-yy.pdf` sendo `xx` e `yy` os *usernames* dos componentes do grupo deve ser enviado por e-mail para `roberto@inf.ufpr.br`;
3. PLÁGIO NÃO SERÁ TOLERADO. É interessante que os alunos conversem sobre o projeto, mas cada grupo deve idealizar e avaliar seu próprio projeto.

Produtos

1. Relatório em papel A4, com letras em 11 pontos, espaço simples, formatação simples, contendo os nomes dos componentes do grupo, e as conclusões de projeto, apontando o(s) melhore(s) projeto(s) e justificando claramente as escolhas. Seu relatório deve conter os gráficos com os resultados das simulações;
2. presença dos membros do grupo na data e hora marcadas para a apresentação.

Sugestões

1. Assegure-se de que entendeu a especificação antes de iniciar o projeto das caches;
2. as simulações geram enormes conjuntos de dados; planeje os experimentos e a coleta e tabulação dos resultados antes de disparar os *scripts* com as simulações.
3. projetistas iniciantes vão atrás das menores taxas de faltas e ignoram a especificação;
4. executar as simulações é demorado, mas é o primeiro passo do processo, e demanda nada além de força bruta;
5. a avaliação das alternativas é o que demanda cérebro. Cérebro = $.

Área, tempo de acesso, energia por referência

Os engenheiros de projeto fornecem os gráficos das próximas páginas para auxiliar na escolha do melhor projeto da cache de dados. O simulador empregado para estas simulações é o CACTI[1].

As capacidade simuladas são 1, 4, 16 e 64 Kbytes, as associatividades 1, 2, 4 e 8, e os tamanhos de bloco 8, 16, 32, 64 e 128 bytes. O primeiro conjunto de valores relaciona capacidade, associatividade e tamanho de bloco com a área resultante do projeto, em mm^2, e é mostrado na Figura 1.

O segundo conjunto de valores relaciona capacidade, associatividade e tamanho de bloco com o tempo de acesso resultante, em ns, e são mostrados nas Figura 2 e Figura 3. Note que o tempo de acesso nem sempre cresce linearmente com a capacidade, tal como área ou energia, porque o simulador escolhe a melhor geometria para otimizar o tempo de acesso. Todas as capacidades entre 1 K e 64 Kbytes foram simuladas para o tempo de acesso.

O terceiro conjunto de valores relaciona capacidade, associatividade e tamanho de bloco com a energia dispendida em cada acesso à cache, em nJ, e é mostrado na Figura 4.

Histórico das Revisões:
17out especificação publicada.

[1] *CACTI 3.0: An Integrated Cache Timing, Power, and Area Model*, Premkishore Shivakumar and Norman P Jouppi, Tech Report WRL-TR-2001.2, aug 2001, Compaq Western Research Lab.

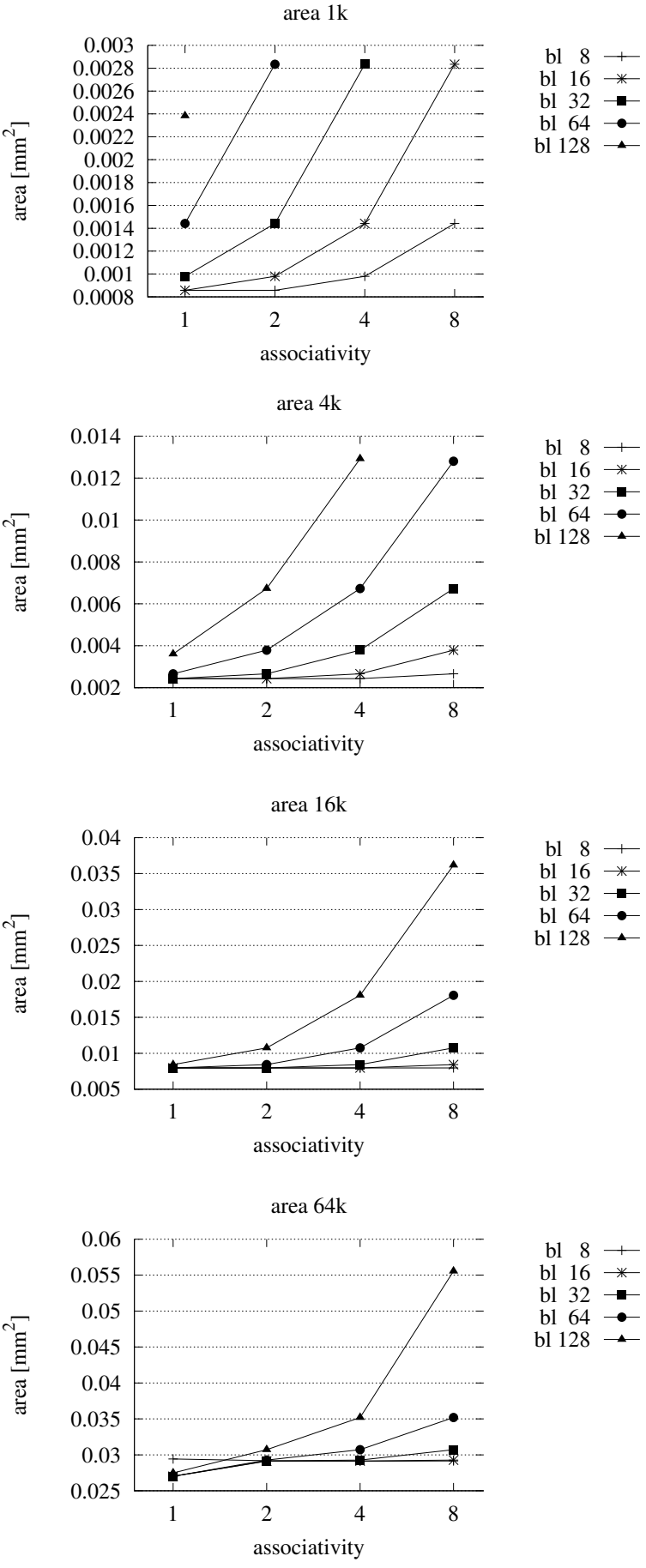

Figura 1: Área X capacidade X associatividade X tamanho de bloco.

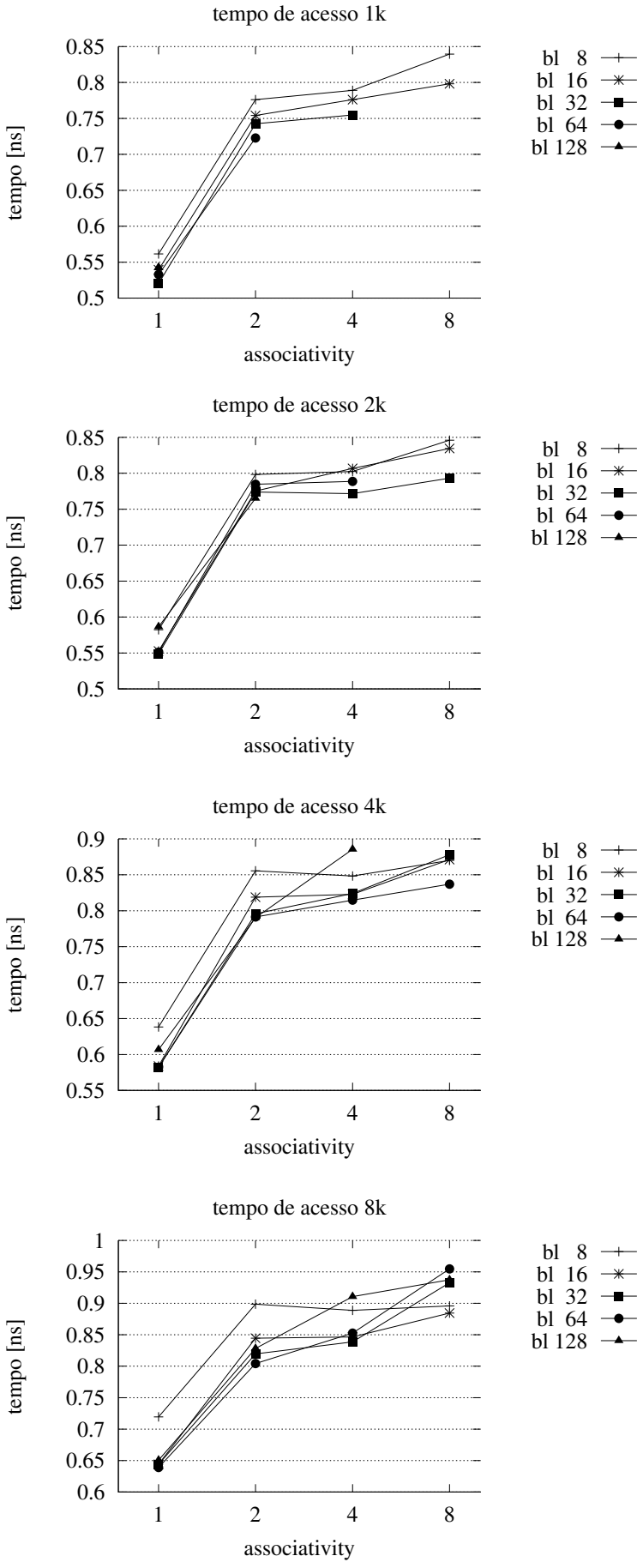

Figura 2: Tempo de acesso X capacidade X associatividade X tamanho de bloco (i).

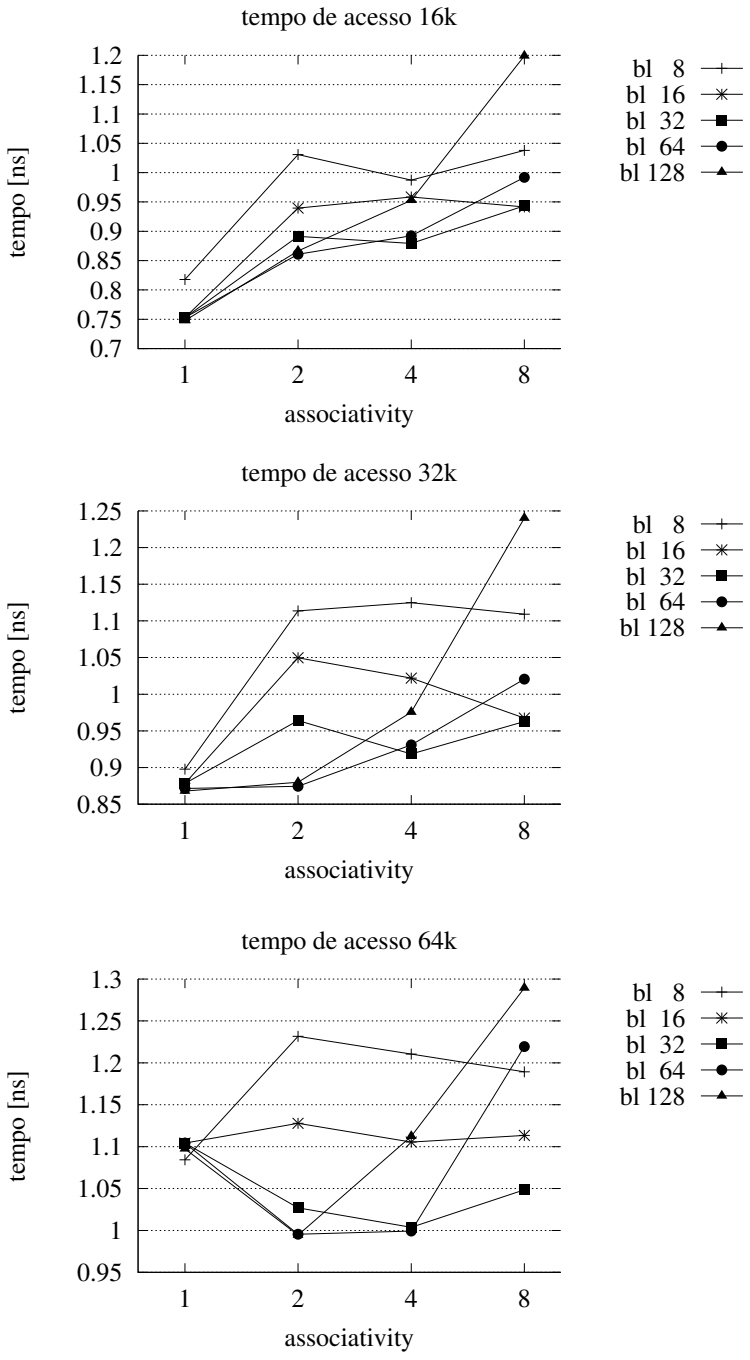

Figura 3: Tempo de acesso X capacidade X associatividade X tamanho de bloco (ii).

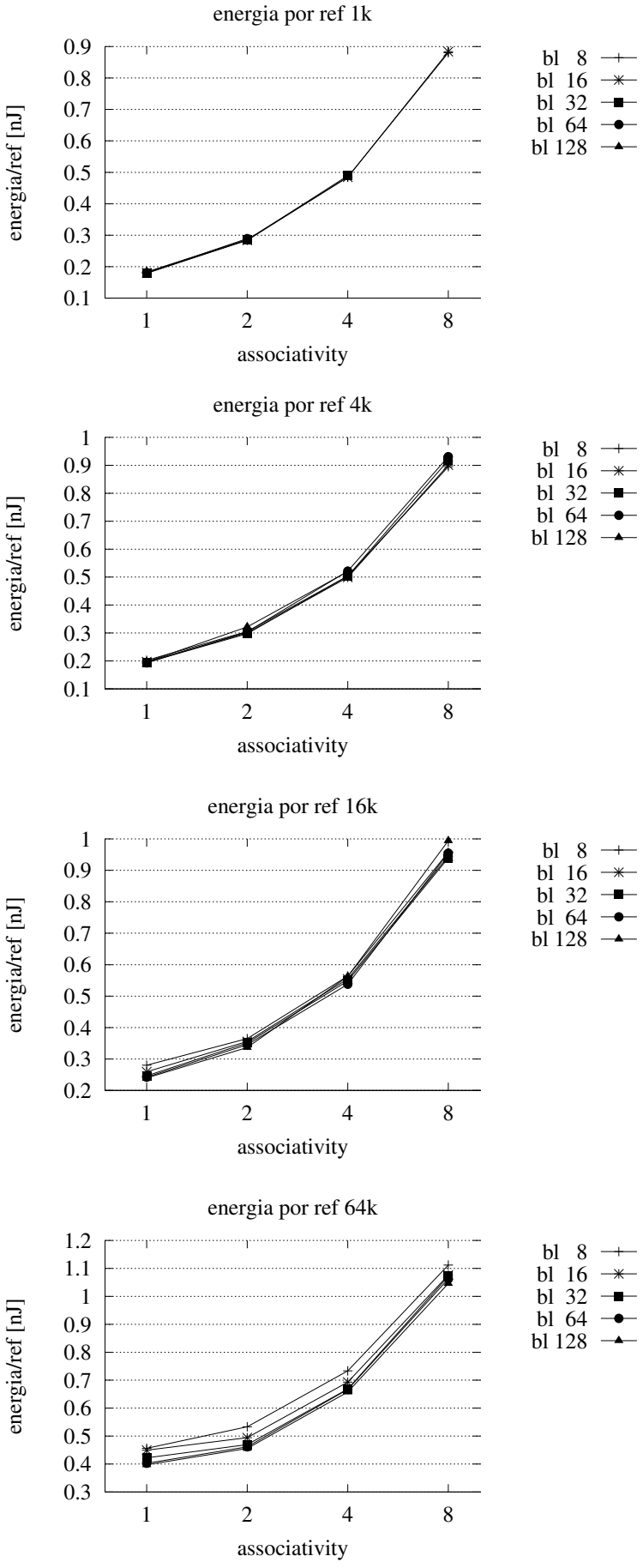

Figura 4: Energia/referência X capacidade X associatividade X tamanho de bloco.