Energy aware Heterogeneous OSEP

Cássio H. V. Forte[1], Aleardo Manacero[1], Renata S. Lobato[1], Roberta Spolon[2]

[1]Departamento de Ciências de Computação e Estatística
Universidade Estadual Paulista "Júlio de Mesquita Filho" - UNESP
São José do Rio Preto – SP – Brasil

[2]Departamento de Computação
Universidade Estadual Paulista "Júlio de Mesquita Filho" - UNESP
Bauru – SP – Brasil

cassiohvforte@gmail.com,{aleardo,renata}@ibilce.unesp.br

Abstract. *Advances in the processing power of large datacenters implied in large growth of energy consumption by them. In recent years there is a search for solutions to reduce this consumption, especially in distributed systems, such as computing grids. One option for such systems is to perform the task's allocation to minimize the total power consumption, constrained to performance requirements. In this paper we present a new scheduling algorithm for grid computing, that controls energy consumption, guided by a fairness criteria based in resource ownership.*

Resumo. *Avanços no poder de processamento de grandes centros são acompanhados pelo aumento significativo no consumo de energia deles. Nos últimos anos buscou-se soluções para reduzir esse consumo, principalmente em ambientes distribuídos, como grades computacionais. Uma alternativa para esses sistemas é fazer a alocação das tarefas minimizando o consumo total, sem perda indesejável de desempenho. O objetivo deste trabalho é propor um novo algoritmo de escalonamento para grades computacionais, que controla o consumo de energia a partir de um critério de justiça baseado na posse dos recursos.*

1. Introdução

Nos últimos anos a preocupação com consumo de energia atingiu grandes *datacenters*, responsáveis por 3% do consumo mundial de energia elétrica gerando cerca de 2% da emissão global de gases do efeito estufa[1]. Isso é muito relevante ao uso de *datacenters* distribuídos, que formam grades computacionais, tanto com gerenciamento proprietário como as grades cooperativas, em que os recursos pertencem a diferentes proprietários.

Para grades cooperativas a alocação de tarefas pode ser feita atendendo critérios diversos, como os baseados em justiça segundo a propriedade de recursos. Neles se busca garantir que usuários que ofereçam recursos para o sistema usem sua parte sempre que quiserem. Nesse sentido Falavinha propôs o algoritmo OSEP (*Owner Share Enforcement Policy*) [Falavinha et al. 2009], restrito a ambientes homogêneos. Forte [Forte et al. 2016] estendeu o algoritmo para grades heterogêneas, dando origem ao algoritmo HOSEP.

[1]Tom Bawden, em http://www.independent.co.uk/environment/global-warming-data-centres-to-consume-three-times-as-much-energy-in-next-decade-experts-warn-a6830086.html

Neste trabalho se faz uma adaptação do critério de justiça baseada em propriedade para atender restrições de consumo de energia, que é o algoritmo EHOSEP (*Energy aware Heterogeneous OSEP*).

2. Trabalhos relacionados

O tratamento do consumo de energia em sistemas distribuídos pode ser feito pelo escalonamento de tarefas ou por ações na infraestrutura. No escalonamento temos trabalhos como os de Etinski [Etinski et al. 2010], que reduz o consumo pela diminuição da velocidade da CPU ou Liu [Liu et al. 2008], que trata da alocação para reduzir tráfego de dados.

Em infraestrutura se busca balancear a carga para reduzir o calor gerado, elevar a eficiência ou diminuir o consumo da infraestrutura de resfriamento e alimentação dos nós [Patil and Chaudhary 2013, Chen et al. 2011]. Do ponto de vista econômico se pode ainda explorar variações nas tarifas cobradas pela energia elétrica [Aikema et al. 2011].

3. Formulação do EHOSEP

O algoritmo proposto é baseado nos algoritmos OSEP e HOSEP, que buscam justiça no atendimento a partir da quantidade de recursos que pertence a cada usuário (OSEP), ou do poder computacional oferecido por esses recursos (porção de usuário) (HOSEP). Com eles se busca garantir que cada usuário seja atendido por um conjunto de máquinas que ofereça poder computacional maior ou igual ao oferecido por eles à grade. Neles se considera tarefas do tipo *bag of tasks*, atendendo-se primeiro o usuário com maior defasagem de uso. No HOSEP essa defasagem é dada pelo diferencial de poder computacional (em MFlops) entre máquinas oferecidas e recebidas ($DifPoder_i = \frac{Alocado_i - Porcao_i}{Porcao_i}$).

3.1. EHOSEP

Os critérios do HOSEP não consideram consumo de energia, que passa a ser tratado no EHOSEP (*Energy aware Heterogeneous OSEP*), que além de considerar a porção de máquinas de cada usuário, estabelece também um limite de consumo definido por ele. Esse limite é dado como porcentagem do consumo total das máquinas que a ele pertencem, não devendo ser ultrapassado. Assim, o consumo de energia pode ser visto como parâmetro de nível de serviço contratado.

No algoritmo EHOSEP se busca atender ao critério de justiça do HOSEP enquanto o limite de consumo permitir. Isso leva a modificações no critério de justiça e ao algoritmo de alocação, visto adiante. O EHOSEP também procura estimular os usuários a oferecerem recursos eficientes energeticamente e, para isso, utiliza a métrica CSP (Coeficiente Sistema/Porção) dado por $CSP_i = \frac{ConsumoPorcao_i}{ConsumoSistema} \times \frac{PoderSistema}{PoderPorcao_i}$.

Nesse algoritmo um usuário x, com limite de consumo $Limite_x$ e com consumo atual $Consumo_x$, é atendido com justiça se uma das condições a seguir for satisfeita:

i. esteja com poder computacional superior ao oferecido ($DifPoder_x \geq 0$);
ii. exceda seu limite de consumo se receber nova máquina, ou seja, $DifPoder_x < 0 \wedge ((Consumo_x = Limite_x) \vee (\forall$ máquina $y \Rightarrow Energia_y + Consumo_x > Limite_x))$;
iii. viole o critério de preempção ao receber nova máquina, ou seja, caso $DifPoder_x < 0 \wedge (Consumo_x < Limite_x) \ \wedge \ (\forall$ usuário $i \mid DifPoder_i > 0$, toda máquina j alocada para i é tal que $(PR_Preemp_{i,j} < 0) \wedge (limite_x \geq limite_i)$;

Tabela 1. Dados sobre porções dos usuários.

Usuário	Número Máquinas	Poder (GFlops)	Consumo (W)	Eficiência (GFlops/W)
U1	16 8	133,47 108	220 191	0,5942
U2	12 8	84,53 95,49	99 113,5	0,8484
U3	6 6	32,43 17,38	134 118	0,1976
U4	8	42,25	67	0,6306

```
Ordena usuários em ordem crescente de DifPoder e faça u_i = primeiro da lista;
while Existir usuário u_i habilitado para atendimento do
    if Consumo_i ≤ Limite_i ∧ existe demanda não atendida de u_i then
        if ∃ máquina livre cujo consumo não faça exceder o limite then
            Aloca menor tarefa de u_i para máquina livre que consumir menos
              energia para executá-la;
        else
            if ∃ u_z com DifPoder > 0, tal que CSP_z × Consumo_z seja o maior,
              e que use máquina M_y cujo consumo não faça exceder o limite then
                if tarefa executada em M_y puder sofrer preempção causando menor
                  desperdício de processamento then
                    if se depois da preempção u_z não tiver falta ou tiver e
                      limite_i < limite_z then
                        Realiza preempção da tarefa t_y do usuário u_z;
                        Aloca M_y para a menor tarefa do usuário u_i;
                    end
                end
            end
        end
        Atualiza dados dos usuários e tarefas, e encerra o algoritmo;
    end
    Faça u_i = próximo da lista;
end
```

4. Testes e Resultados

Os testes foram feitos com o simulador iSPD, com um modelo de grade com 64 máquinas divididas entre os usuários U1, U2, U3, e U4 (Tabela 1). As tarefas submetidas têm tamanhos variando de 10 a 60 minutos de execução numa máquina de poder equivalente ao poder médio do sistema.

Por restrições de espaço apresentamos aqui resultados de apenas um caso de teste, em que os usuários U2, U3 e U4 têm limites acima de suas porções (para que ocupem todo o sistema) e o usuário U1 tem seu limite variando de 80% a 182,09% do consumo de suas máquinas. A satisfação dos usuários (Figura 1), calculada pela equação 1, mostra que o algoritmo deixa mais satisfeitos os usuários que entregam mais recursos e que tentam consumir menos energia (satisfação de U1 cai quando seu limite passa de 150%)

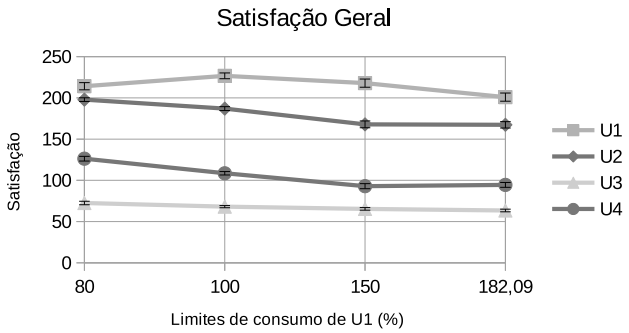

Figura 1. Resultado dos testes, média das satisfações dos usuários

$$S_i = 100 \times \sum_{t=1}^{N} \frac{TempoIdealTar_t}{TempoRealTar_t} \times \frac{ConMaxPorcao}{ConsTotalPorcao} \times \frac{ConsTotalSist - ConsTotal_i}{Limite_i \times TempoTotalExec} \quad (1)$$

5. Considerações Finais

O algoritmo EHOSEP, apresentado neste trabalho, traz uma abordagem de justiça baseada no consumo de energia e de oferta de poder computacional. Com ele se atende melhor usuários que buscam consumir menos energia do que oferecem ao sistema, tornando o conjunto mais eficiente energeticamente.

Referências

Aikema, D., Kiddle, C., and Simmonds, R. (2011). Energy-cost-aware scheduling of HPC workloads. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pages 1–7. IEEE.

Chen, H., Song, M., Song, J., Gavrilovska, A., and Schwan, K. (2011). HEaRS: A hierarchical energy-aware resource scheduler for virtualized data centers. In *Cluster Computing (CLUSTER), 2011 IEEE International Conference on*, pages 508–512. IEEE.

Etinski, M., Corbalan, J., Labarta, J., and Valero, M. (2010). BSLD threshold driven power management policy for hpc centers. In *IEEE International Symposium on Parallel & Distributed Processing, (IPDPSW)*, pages 1–8. IEEE.

Falavinha, J., Manacero, A., Livny, M., and Bradley, D. (2009). The owner share scheduler for a distributed system. In *Intl Conf on Parallel Processing Workshops, ICPPW'09*, pages 298–305.

Forte, C., Manacero, A., Lobato, R., Spolon, R., and Falavinha-Jr, J. (2016). Política de escalonamento owner-share para sistemas de grades heterogeneas. In *Anais do XVII Simp. em Sist. Computacionais de Alto Desempenho - WSCAD*, pages 206–217.

Liu, C., Qin, X., Kulkarni, S., Wang, C., Li, S., Manzanares, A., and Baskiyar, S. (2008). Distributed energy-efficient scheduling for data-intensive applications with deadline constraints on data grids. In *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, pages 26–33. IEEE.

Patil, V. A. and Chaudhary, V. (2013). Rack aware scheduling in HPC data centers: An energy conservation strategy. *Cluster Computing*, 16(3):559–573.