8
Duração x Custo

8.1
Fronteiras de Eficiência

As aplicações em ambiente de computação na nuvem podem proporcionar significativa economia de tempo em virtude da possibilidade de paralelização do processamento em diversos nós distintos (instâncias). Em um mundo ideal, quanto maior o número de nós da aplicação, mais rápido a conclusão do processamento. Na prática, essa pode não ser a melhor estratégia, pois quanto maior o número de nós, maior o tempo gasto com o gerenciamento da paralelização e o custo total do processamento.

Em geral, as aplicações nestes ambientes, como na prova de conceito apresentada no capítulo 7, restringem-se à demonstração da efetiva redução do tempo de processamento como resultado da paralelização das tarefas, e dos custos uma vez que só é necessário pagar pelos recursos utilizados. No entanto, a relação entre os recursos utilizados (número de nós, duração do processamento, custo total, etc.) e as preferências do usuário, de forma a maximizar benefícios, é muito pouco explorada.

Neste capítulo apresentamos a análise realizada em uma gama particular de aplicações e o modelo construído para que as preferências do usuário, em relação à duração e custo total, sejam determinantes na configuração da sua execução, em especial, na definição do número de nós e na estratégia de divisão do processamento entre os mesmos.

O modelo proposto é genérico, portanto, em cada implantação do McCloud as fronteiras de eficiência e características devem ser analisadas para aderência ao modelo.

8.2
Notação da Modelagem

Letra	Descrição	Unidade
N	Número total de realizações a serem executadas	Inteiro
I	Tempo de computação estimado de uma única interação	Horas
P	Tempo necessário ao gerenciamento de cada partição/tarefa	Horas
D	Tempo de implantação e exclusão de uma instância	Horas
X	Número de partições por instância	Inteiro
V	Valor da hora de computação de uma instância	Dólar
W	Número de instâncias	Inteiro
T	Tempo decorrido no relógio do início ao fim da computação	Horas
C	Custo do tempo de computação total	Dólar
U	Preferência do usuário (em relação ao W ótimo)	Percentual

Tabela 4 – Notação da modelagem

Adotaremos a notação da Tabela 4 para facilitar a modelagem da otimização. P contempla a entrada e saída de dados de cada partição, o tempo de divisão da computação para uma única partição, e a posterior junção dos resultados desta única partição com o resultado final anterior.

As variáveis N, I, P, D e V podem ser consideradas constantes para a construção do modelo, visto que:

a) N é fornecida obrigatoriamente como parâmetro de entrada, visto que o processamento tem um número de realizações (amostra) previamente definido;

b) I pode ser calculado pela média da duração da execução de uma quantidade muito pequena de realizações e, no caso daquelas perfeitamente simétricas, ele consiste na duração de uma única interação;

c) P e D podem ser estimados por simples observação, e independem do que está sendo processado;

d) V é definido pela regra comercial da plataforma.

8.3
Modelagem Proposta

A duração (T) é modelada conforme equação E1, onde o tempo linear de computação (N x I) é dividido pelas instâncias (W), somando o tempo gasto no gerenciamento de cada partição (P x X) e o tempo de implantar e excluir todas as instâncias em paralelo (D).

$$\text{(E1)}\ T = \frac{(N.I)}{W} + (P.X).W + D$$

As análises que se seguem são válidas somente se o volume de realizações é consideravelmente grande, ou seja, se E2 é verdadeiro. Essa premissa é na teoria

sempre satisfeita, pois a gama de aplicações de interesse possui esforço computacional total significativo.

(E2) $(N.I) > (P.X)$

A partir de E1 podemos concluir que a duração máxima (T_{max}) ocorre quando utilizamos apenas uma instância (W=1) e descrevê-la conforme E3.

(E3) $T_{\max} = (N.I) + (P.X) + D$

Para encontrar o número de instâncias (W) que permite a duração mínima (T_{min}), ou seja, o número de instâncias ótimo (W_1), é necessário igualar a zero a primeira derivada de E1. Como o número de instâncias precisa ser um inteiro positivo, temos E4.

(E4) $T^{'} = -\frac{(N.I)}{W_1^{\,2}}. + (P.X) = 0 \therefore W_1 = \sqrt{\frac{N.I}{P.X}} \uparrow$

Utilizando apenas uma partição por instância (X=1) e calculando o número de instâncias ótimo (W_1), obtem-se a duração mínima (T_{min}), conforme E5.

(E5)
$$X = 1 \therefore W_1 = \sqrt{\frac{(N.I)}{(P.1)}} \uparrow \therefore$$
$$T_{\min} = \frac{(N.I)}{W_1} + (P.1).W_1 + D$$

Dessa forma, fica evidente que aumentar indiscriminadamente o número de instâncias (W) não garante uma duração menor do processamento, pois abaixo ou acima do número ótimo de instâncias (W_1) temos uma duração maior.

Na duração mínima calculada acima adotamos apenas uma partição/tarefa por instancia. Na prática, se ocorrer alguma falha no processamento de uma instância, a duração total se multiplicará pelo número de falhas sequenciais, pois somente após esperar o tempo que seria necessário para a instância processar (*timeout*) se torna possível identificar a interrupção ou erro na partição e, então, para daí reiniciar o processamento.

É prudente dividir o processamento de uma instância/tarefa de modo que eventuais falhas causem atrasos mínimos. A duração total na ocorrência de erros de acordo com a quantidade de erros consecutivos (E), pode ser expressa conforme E6.

$$T_{erro} = T_{\min} + \frac{T_{\min}.E}{X} = (1 + \frac{E}{X}).T_{\min} \therefore$$

(E6)

$$T_{erro} = \left(1 + \frac{E}{X}\right).\left(\frac{(N.I)}{W} + (P.X).W + D\right)$$

Ao tentarmos encontrar o número ótimo de instâncias (W_2) que permita uma duração mínima com erro, igualando a zero a primeira derivada da duração com erro (T_{erro}), obtido em E6, chegamos à relação de W demonstrada em E7, ou seja, W_2=W_1. Conlui-se, assim, que a escolha de W independe de E (erro).

$$T_{erro}{}' = \left(-\frac{(N.I)}{W_2{}^2} + P.X\right).\left(1 + \frac{E}{X}\right) = 0$$

(E7)

$$\therefore W_2 = \sqrt{\frac{N.I}{P.X}} \uparrow$$

Considerando que o número ótimo de instâncias (W_2) depende exclusivamente do número de partições/tarefas por instâncias (X) e outras constantes, conforme demonstrado em E7, podemos analisar o limite da duração com erro (T_{erro}) quando o número de partições tende a ser infinito (X), conforme E8.

$$T_{erro} = \left(1 + \frac{E}{X}\right).\left(\frac{N.I}{\sqrt{\frac{N.I}{P.X}}} + P.X.\sqrt{\frac{N.I}{P.X}} + D\right)$$

(E8)

$$\therefore \lim_{X \to \infty} T_{erro}(X) = \infty$$

Considerando que nessas condições o tempo tende a infinito, podemos concluir que, diferente da suposição anterior, o aumento do número de partições não diminui a duração total na ocorrência de erro, porque impõe um esforço adicional de gerenciamento, maior que o custo de eventual erro. Portanto, nosso modelo de duração ótima (T_o) fica conforme E9, visto que o número de partições ótimo (X_o) é igual a 1 (X_o =1).

(E9) $W_o = \sqrt{\frac{N.I}{P}} \uparrow \therefore T_o = \frac{N.I}{W_o} + P.W_o + D$

O custo base (C_b), que desconsidera as particularidades de cobrança de cada plataforma, pode ser calculado multiplicando a duração (T) pelo valor (V) e pelo número de instâncias (W), conforme E10.

(E10) $C_b = T.W.V$

A computação na plataforma Microsoft Windows Azure, adotada pelo McCloud, é cobrada de acordo com as horas "distintas" do relógio em que a instância esteve alocada, ou seja, do momento em que a instância iniciou sua implantação até o momento em que a mesma foi completamente excluída, independente de paralisações nestes intervalos. Frações são consideradas como horas completas (cheias).

Além disso, se a instância é excluída e reimplantada, uma nova hora completa é cobrada. Por exemplo, se for implantada uma instância às 10h50min e excluída às 11h10min, serão cobradas duas horas completas, a primeira referente ao período de 10h50min até 11h e a segunda de 11h até 11h10min.

O custo de cada hora é apresentado na Tabela 5. Observe que o tamanho/custo da instância é apenas um multiplicador em relação à instância pequena. No contexto deste trabalho utilizamos apenas a instância pequena e, consequentemente, a constante V será o valor da hora desta instância.

Size	CPU	Memory	Storage	I/O Perf.	Hour
Small	1.6 GHz	1.75 GB	225 GB	Moderate	USD$ 0,12
Medium	2 x 1.6 GHz	3.5 GB	490 GB	High	USD$ 0,24
Large	4 x 1.6 GHz	7 GB	1,000 GB	High	USD$ 0,48
Extra Large	8 x 1.6 GHz	14 GB	2,040 GB	High	USD$ 0,96

Tabela 5 – Custo plataforma Windows Azure

O custo de armazenamento não será analisado nesta otimização, pois não está, a priori, diretamente relacionado à estratégia de paralelização.

Considerando o exposto sobre a plataforma selecionada, para cálculo de uma previsão do custo de processamento (C) é necessário arredondar a duração calculada (T) para o inteiro imediatamente acima e, ainda, somar uma hora para considerar como completa a hora do relógio de término, conforme E11.

(E11) $C = (T \uparrow +1).W.V$

Analisado a duração mínima (E9), a duração máxima (E3), o número ótimo de instâncias (W_o) e o custo (E11) se verifica que para obtermos uma duração

menor o valor será maior. Da mesma forma, para obtermos um valor menor é preciso estar disposto a aguardar uma maior duração.

Dessa forma, é importante entender as preferências do usuário, para que o ambiente seja configurado de forma a priorizar suas preferências no equilíbrio entre duração e custo, maximizando benefícios.

Como o custo e a duração modelados são apenas para obtermos previsões, seria inadequado modelarmos as preferências em função direta destas previsões, pois poderíamos criar falsas expectativas no usuário. Dessa forma, escolhemos modelar a preferência do usuário em função do número de instâncias (W_o), ou seja, escolhendo um percentual de preferência (U) entre uma única instância (W=1) e o número ótimo de instâncias ($W=W_o$), conforme E12.

$$(E12)\ W_p = \left((\sqrt{\frac{N.I}{P}} - 1) * U + 1 \right) \uparrow$$

Observe que U varia de 0% a 100% portanto, W_p varia de 1 a W_o, consequentemente, o usuário pode fazer sua seleção desde o menor custo e maior duração até o maior custo e menor duração.

Saliente-se que o modelo proposto considera uma avaliação incremental de nós a serem provisionados no início de cada simulação, e desprovisionados ao término.

Na Figura 33 apresentamos uma ilustração da variação do custo e da duração em função da preferência do usuário (U), ou melhor, do total de nós (W). Para facilitar a leitura do gráfico, optamos por um cenário onde W_o=100, para U poder ser lido no gráfico como igual a W.

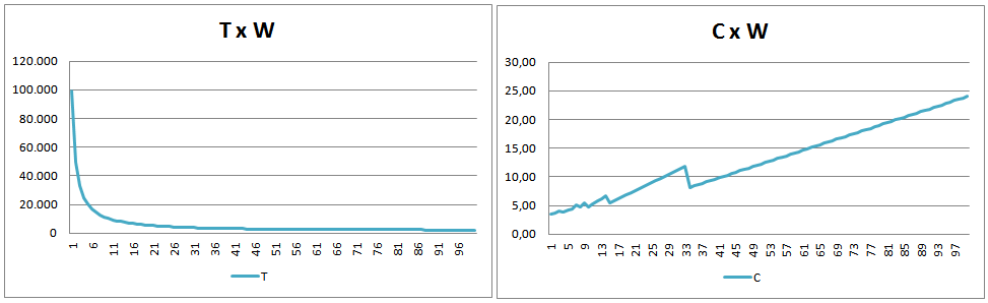

Figura 33 – Gráfico da influência de W na duração e custo

Na prática, ao variarmos U crescentemente (eixo horizontal), a duração possui a forma praticamente exponencial decrescente e o custo possui forma praticamente linear crescente.

Cabe ao desenvolvedor, frente aos requisitos do negócio, decidir qual o valor de U adequado. Por exemplo, quando o negócio demanda uma resposta imediata, independente do custo, deve ser utilizado U=100%. Por outro lado, quando não se deseja gastar muito, pode-se, por exemplo, utilizar U=40%, balanceando um ganho significativo na duração sem alto acréscimo de custo.

8.4 Aplicação Teórica

Para aplicação da modelagem proposta na prova de conceito do capítulo 7:

a) Estimamos I através da divisão do tempo de processamento da maior amostra (1.696,69 segundos) pelo tamanho desta amostra (10.0000.000.000 realizações) e multiplicamos pelo número de instâncias (19), que arredondando para baixo resultou em 3.10^{-7} segundos.
b) A observação de D foi de 7 minutos (420 segundos).
c) O P foi estimado através da média de todos os experimentos (*split* e *merge*), que arredondando para cima, ficou em 0,30 segundos.

Repare que fizemos os arredondamentos com essa estratégia para compensar parte do esforço de P que está embutido em I. Além disso, é importante reforçar que estamos apenas inferindo estimativas.

A tabela apresenta a seguir demonstra a aplicação da modelagem.

N	Número total de interações a serem computadas [inteiro]	100.000.000.000
I	Tempo de computação de uma única interação [segundos]	0,000000300000
P	Tempo necessário ao gerenciamento de cada partição [segundos]	0,3
D	Tempo de implantação (deploy) e exclusão de uma instância [segundos]	420
V	Valor da hora de relógio da computação de uma instância [USD$]	0,12

		W=1	W=19	Wo	Wp	W+10
Válido?	O modelo aplica-se somente quando N é suficientemente grande (E2)	Sim	Sim	Sim	Sim	Sim
U	Percentual de preferência do usuário [percentual]	-	-	100%	80%	110%
W	Número de instâncias [inteiro] (E4/E12)	1	19	317	254	348
T	Tempo decorrido no relógio do início ao fim da computação [segundos] (E1)	30.420	2.005	610	614	611
C	Custo do tempo de computação total [USD$]	1,2	4,56	76,08	60,96	83,52

Tabela 6 – Comparativo de tempo e custo em diferentes cenários

Nessa tabela apresentamos cinco cenários distintos:

a) Apenas um nó (W=1);
b) Com 19 nós definidas de forma arbitraria (W=19);
c) Número ótimo de nós (W_o) calculado com o modelo, U=100%;
d) Nó equivalente a 80% do número ótimo (W_p), ou seja, com U=80%;
e) Nós teóricos 110% do número ótimo (W_p), ou seja, com U=110%.

Com W dos cenários C e D temos teoricamente uma duração mais eficiente (menor) que com o W dos casos A e B. Além disso, o caso E ilustra que a partir do ponto ótimo o aumento de instâncias não resulta em redução de duração, ao contrário, aumenta o custo.

8.5 Implementação na Prova de Conceito

Para ilustrar os benefícios desta modelagem, optamos pela aplicação na prova de conceito. Primeiramente com 100.000.000.000 realizações e preferência do usuário em 31% (U) para demonstrar que a simulação pode ser realizada de forma mais rápida a partir da utilização do modelo. Posteriormente com 1.000.000.000.000 realizações e preferência do usuário em 9,8% (U) para reforçar que o tempo apurado na prática é na ordem de grandeza do previsto pelo modelo. Aproveitamos para destacar que devido ao modelo de cobrança do Azure por hora cheia, o custo de ambas as simulações foram de USD$12,15. Lembrando que para adequada comparação com os experimentos anteriores, esse tempo desconsiderará D e os nós serão previamente levantados. Sendo assim, teremos no ponto de extensão *optimization* a configuração de *ntasks* igual a 99. Ao executar nessas condições obtivemos os resultados abaixo:

MODELO DE OTIMIZAÇÃO PERFORMANCE																
Entrada			Configuração			Tempo (segundos)				Blob		CUSTO	Modelo		Comparação	
#	N	Precisão	W	Tarefas	Tar./W	Split	Process	Merge	Total	Interno	Saída	USD$	U	Previsão	Tempo	Razão
10	100.000.000.000	5	99	99	1	2,36	339,31	1,22	**342,88**	990 Byte	13 Byte	**12,15**	31%	333,00	98.482,969	287,22
11	1.000.000.000.000	5	99	99	1	2,09	3278,11	1,20	**3281,40**	990 Byte	13 Byte	**12,15**	9,8%	3.060,00	984.829,690	300,12

Tabela 7 – Resultado da aplicação do modelo proposto

Os resultados obtidos nesse experimento demonstram a vantagem de possuir um modelo como este, para melhor configuração e previsão das simulações.

Projetando o tempo para execução desta mesma amostra com um trilhão de realizações com computação tradicional, a duração seria na ordem de 11,4 dias (10 x 98.482,969s da Tabela 6). Considerando a razão desta duração com o resultado obtido na tabela acima com 99 nós, temos uma simulação 300 vezes mais rápida.

8.6
Generalização do Modelo

Diante do exposto, propomos uma implementação genérica utilizando o modelo (E12) no ponto de extensão *optimization*, conforme figura abaixo.

```
public void optimization(double n, string codein, string codeout,
    out double ninstancesadded, out double ntasks, out int timeoutInSeconds)
{
    double i = 0.0000003;
    double u = 0.18;
    double p = 0.3;
    double w = ((Math.Sqrt(n * i / p) - 1) * u + 1);
    ninstancesadded = Decimal.ToInt32(Decimal.Truncate((decimal)w));
    ntasks = w;
    timeoutInSeconds = 2 * 60 * 60;
}
```

Figura 34 – Implementação genérica proposta do *optimazation*

Os parâmetros constantes deste exemplo são I = 0,000003s, U = 18% e P = 0,3s. Estes podem ser ajustados com os valores desejados para cada domínio.

Repare que com base nas constantes é feito o cálculo com o modelo (E12) do número de instâncias a serem levantadas adicionalmente para execução da simulação (*ninstancesadded*) e do número de tarefas na qual a simulação será dividida (*ntasks*).