

MiniEP 4 - MAC0219

Bruno Daiki YAMADA N° 9348780

1 de Maio de 2018

1 Métodos

Para resumir nosso dataset de tamanho $n = 30$, representado por x , utilizaremos a média amostral:

$$E[x] = \frac{\sum_{i \in x} i}{n} \quad (1)$$

Para resumir a dispersão dos dados, utilizaremos a variância amostral:

$$S_x^2 = \frac{n}{n-1} \left(E[x^2] - (E[x])^2 \right) \quad (2)$$

A fração $\frac{n}{n-1}$ é a correção de Bessel, que permite que S_x^2 seja um estimador não enviesado da variância da população.

Para provar que em média um algoritmo de fato é mais lento do que outro, utilizaremos o intervalo de confiança:

$$IC(\mu_x, 90\%) = \left[E[x] - \frac{S_x}{30} \cdot 1.6991; E[x] + \frac{S_x}{30} \cdot 1.6991 \right] \quad (3)$$

No qual μ_x é a média da população x , M_x média amostral do experimento, S_x desvio padrão amostral. Para isso, assumimos que a distribuição de M_x é descrita pela distribuição de probabilidade t-Student com 29 graus de liberdade, pois o tamanho da amostra é 30 e a variância da população é desconhecida.

Se os dois intervalos de confiança têm intersecção vazia, consideramos que temos evidência estatística necessária para afirmar que os dois valores são diferentes.

O programa foi modificado para obter a média amostral e a média quadrática amostral dos tempos de execução.

2 Resultados

2.1 Tempo de execução

Executamos o teste com 30 threads e 30000000 total time primeiro:

Tabela 1: Resultados com 30 threads e 30000000 total time				
Algoritmo	$E[x]$	$E[x^2]$	$S_x/301.6991$	$IC(\mu_x, 90\%)$
Bakery	264423070	2180039196029642419	49248602	[215174468; 313671672]
Gate	17030225	10081044202479247	3354704	[13675521; 20384929]

Observamos primeiramente que apesar do intervalo de confiança de cada algoritmo ser *relativamente* largo, como há uma ordem de grandeza de diferença entre os tempos de execução, não há intersecção entre os intervalos de confiança, e concluímos que de fato o algoritmo Bakery tem um tempo de execução estatisticamente pior do que Gate.

Agora executamos o mesmo teste para 10 threads e 30000000 total time:

Tabela 2: Resultados com 10 threads e 30000000 total time				
Algoritmo	$E[x]$	$E[x^2]$	$S_x/301.6991$	$IC(\mu_x, 90\%)$
Bakery	68894966	147644738245192884	12816037	[56078929; 81711003]
Gate	10921688	3601019432904782	2000498	[8921190; 12922186]

Diminuindo a concorrência, concluímos que ainda há uma diferença considerável entre os tempos de execução dos algoritmos, ainda sendo estatisticamente diferentes. Entretanto da diferença entre as médias é consideravelmente menor, sendo pelo menos na mesma ordem de grandeza.

2.2 Fairness

Para avaliar a fairness entre as duas soluções do problema, observaremos os valores de desvio padrão do conjunto de valores de acesso a região crítica. Se um algoritmo é justo, esperamos que as diferentes threads tenham quantidade de acessos a região crítica similar, resultando em variância baixa e consequentemente um desvio padrão baixo.

O mesmo teste de intervalo de confiança não pode ser aplicado pois a distribuição de probabilidade de variância não é descrito por uma distribuição de t-Student. Entretanto observaremos exemplos de execuções individuais e avaliaremos a diferença de desvios padrões entre os dois algoritmos nas duas situações descritas anteriormente.

Tabela 3: Resultados com 30 threads e 30000000 total time com scheduler OTHER

Algoritmo	Desvio padrão do número de acessos
Bakery	831.437484
Gate	5474.673295

Tabela 4: Resultados com 10 threads e 30000000 total time com scheduler OTHER

Algoritmo	Desvio padrão do número de acessos
Bakery	260.159355
Gate	2663.055780

Observamos que nos dois casos Bakery teve uma variância menor no desvio padrão do número de acessos a região crítica, confirmando que de fato o algoritmo Bakery permite o acesso a seção crítica de maneira mais justa do que Gate.

Agora observaremos a influência no scheduler na fairness dos algoritmos:

Tabela 5: Resultados com 10 threads e 30000000 total time com scheduler FIFO

Algoritmo	Desvio padrão do número de acessos
Bakery	1.135782
Gate	3424.229708

Tabela 6: Resultados com 10 threads e 30000000 total time com scheduler RR

Algoritmo	Desvio padrão do número de acessos
Bakery	3.269557
Gate	4027.899067

Observamos que o desvio padrão do número de acessos a seção crítica com os scheduler Round Robin e First in first out foram menores do que os scheduler padrão para o algoritmo Bakery. Entretanto, a variância aumentou para esses dois escalonadores para o algoritmo Gate.

2.3 Atomicidade

O bom funcionamento dos algoritmos de exclusividade mútua depende dos comandos `sync_synchronize`, realizando os testes incluídos sem a presença desses comandos resulta na falha da exclusividade

mútua. A razão pela qual esse comando é necessário é a não atomicidade de operações simples, como atribuição. Se uma variável é verificada durante sua atribuição não atômica, valores anômalos podem ser verificados, não podendo garantir o funcionamento do algoritmo e potencialmente perdendo a exclusividade mútua.