Trabalho 09

Disciplina: Programação Paralela **Data: 07 de maio de 2015.** **Prof.:** Fernando Castor

1. Implemente uma trava TAS com *backoff* exponencial, usando a interface `Lock` como base. Sua trava deve funcionar da maneira **mais eficiente possível** e não precisa ser reentrante. Crie um teste que usa essa implementação para proteger o acesso a um objeto contador que possui apenas um método, `incrementa()`, auto-explicativo. Construa um programa onde, durante 2 minutos, 10 threads executam o método `incrementar()` repetidamente. Quantas vezes cada thread conseguiu executar `incrementar()`? Considere que a escolha do objeto a partir do qual cada thread fará o incremento é aleatória. E se forem 50 threads? E 100? E 200? Agora desligue a política de *backoff* que você utilizou. Como o número de execuções foi afetado em cada caso? E se sua política de *backoff* fosse aditiva, ao invés de exponencial? Agora retire o limite de tempo e faça com que cada thread execute o método `incrementar()` 1.000 vezes. Qual o tempo de execução em cada um dos cenários descritos anteriormente? Compare o desempenho da sua trava com a da classe `ReentrantLock` de Java. Modifique sua trava para que ela torne-se uma TTAS e repita os experimentos. Apresente os resultados para todos esses casos e lembre-se que, para experimentos com desempenho, várias execuções são necessárias!

2. Agora implemente uma trava de fila tão eficiente quanto possível. E repita o experimento anterior utilizando-a. Os resultados mudaram? Por quê?

Exercícios do AMP:

- 85
- 86
- 91