

Metodologia PCAM - Atividade 2

Bruno Baldissera Carlotto	10724351
Bruno Gazoni	7585037
Gabriel Eluan Calado	10734453
João Villaça	10724239
Matheus Steigenberg Populim	10734710

Particionamento

Visto que os dados são inseridos pelo stdin na forma de linhas, aumentaremos a eficiência de uso de memória transpondo a matriz e trabalhando os elementos a serem usados em cálculos em linhas ao invés de colunas. As linhas resultantes são independentes, e podemos assim particionar um problema $O(n)$ sequencial em 8 threads, resultando em $O(n/8)$. Evitamos o uso de uma thread por linha para não afetar o speedup de forma negativa, criando threads desnecessariamente e afetando a performance negativamente pelo tempo necessário a mais na própria criação de threads.

Comunicação

O cálculo do desvio padrão depende do cálculo prévio da variância que por sua vez depende da média aritmética, bem como a moda e mediana dependem de ordenação dos elementos de entrada. Estes cálculos devem assim propagar seus resultados de forma apropriada, na ordem apresentada, para evitar redundância de operações e inconsistências. Após o multicast destes resultados, todos os resultados devem ser transmitidos a um vetor de resultado.

Aglomerção

De acordo com as condições explícitas na seção de comunicação, devem ser criadas 3 regiões de paralelismo aglomerando os cálculos: uma tarefa para média harmônica, outra para média aritmética, desvio padrão, variância e coeficiente de variação, e outra para moda e mediana que inclui ordenação prévia. Isso implicará em granulação mais grossa que nesse caso será benéfica para o desempenho. Essas regiões paralelas serão paralelizadas em 8 threads (apesar da natureza n-dimensionnal do problema) definidas estaticamente, de acordo com a justificativa explícita na seção de particionamento. Teremos duas operações auxiliares que serão feitas de forma sequencial: para a checagem da existência da moda, faremos uma distribuição dos elementos em buckets e então conferimos se o número de buckets equivale ao número de elementos, caso sim não há moda e retornamos -1. Para mediana, devemos ordenar os elementos utilizando qsort, que não será paralelizado.

Mapeamento

O número de threads a serem utilizadas do cluster foi definido estaticamente, como justificado na seção de particionamento. Será utilizada a diretiva de omp for para a paralelização da operação em 8 threads, que, devido ao desempenho homogêneo dos nós do cluster, será mapeada de acordo com um algoritmo round-robin. O código deverá ser reconfigurado para consideração em plataformas de desempenho heterogêneo, tendo cuidado em particular com a possibilidade de replanejamento de número de threads caso seja vantajoso para esse sistema criar threads em período menor de tempo, justificando o particionamento em n threads. Da mesma forma, as operações de sort e agrupamento em buckets poderiam ser alocadas em nós especializados, caso fosse possível e desejável.