

Questão: *Descreva em uma página os conceitos de programação funcional necessários para o desenvolvimento de algoritmos distribuídos para trabalhar com grande massa de dados.*

Big Data é um conceito que descreve o grande volume de dados estruturados e não estruturados que são gerados a cada segundo. Quanto mais as coisas mudam, mais as mudanças são capturadas e registradas como dados [1]. Essa grande massa de dados possibilita que se obtenha respostas mais completas e, portanto, mais confiança nos dados e consequentemente, uma abordagem diferente para lidar com os problemas [1][2].

Os dados possuem valor intrínseco, porém não há valor algum até que seja descoberta alguma informação de utilidade. Muitas vezes as técnicas tradicionais de mineração e manuseio de dados não conseguem descobrir os *insights* e o significado dos dados subjacentes [3]. Dados não estruturados, sensíveis a tempo ou simplesmente muito grandes não podem ser processados por mecanismos de bancos de dados relacionais. Esses tipos de dados requerem uma abordagem de processamento diferente que use paralelismo massivo [2][3]. Neste ponto é onde a programação funcional colabora para tornar a análise de dados massivo viável e efetiva.

A principal característica da programação funcional é a ausência de efeitos secundários, que é possível porque os dados são imutáveis, ou seja, dados atribuídos a uma variável (na prática, é apenas um símbolo da definição) [4] não é mais alterado. As suas funções são sem estado, portanto, os valores computados anteriormente não influenciam sua resposta atual, e as funções são geralmente puras, ou seja, não dependem de variável externa e não alteram quaisquer valores externos a elas. Essas funções permitem computar expressões em praticamente qualquer ordem, favorecendo o paralelismo. Também colabora para a otimização da execução do código, pois se o resultado não for utilizado, este não precisa ser computado [4]. Semelhante a uma função matemática, um código funcional tem a sua saída condicionada exclusivamente à sua entrada, de modo que esta última garanta sempre a mesma saída após a execução do mesmo método [5]. A programação funcional também é declarativa, ou seja, o código contém o que precisa fazer e não como ser feito, o que facilita a obtenção dos resultados esperados. Essa característica somada ao fato da linguagem utilizar apenas recursão ao invés de iteração, tornam código ainda mais enxuto e otimizado [4][5].

Juntas, todas essas características facilitam o desenvolvimento de algoritmos distribuídos e, consequentemente, os cálculos realizados. A programação funcional garante que não haverá efeitos colaterais e que o resultado será sempre o mesmo se a entrada for a mesma. Isso facilita a busca por falhas, seja de máquina ou de rede que são muito comuns, e também facilita a execução de testes [4]. As funções, por serem imutáveis, facilitam a distribuição dos cálculos em várias máquinas com o uso do paralelismo. Ao reduzir ou eliminar a mutabilidade, as abordagens funcionais tendem a levar a programas que lidem com paralelismo e escalabilidade naturalmente [5].

O desenvolvimento de frameworks de código aberto, como o Hadoop e Spark, foi essencial para o crescimento de Big Data, porque eles facilitam o trabalho com grande massa de dados através do uso da programação funcional, pois esta possui qualidades que viabilizam o processamento do chamado *Big Data*, provendo análises mais rápidas em grande quantidade de dados através de algoritmos distribuídos.

Bibliografia

- [1] MAYER-SCHÖNBERGER VIKTOR; CUKIER, K. **Big data: a revolution that will transform how we live, work and think**. John Murray, 2017.
- [2] Oracle, "**What is Big Data**". Disponível em:
<<https://www.oracle.com/big-data/guide/what-is-big-data.html>>
- [3] MEMSQL, "**The Resurgence of Scala for Big Data**". Disponível em:
<<http://blog.memsql.com/big-data-scala/>>
- [4] CHIUSANO, P.; BJARNASON RÚNAR, "**Functional programming in Scala**". Manning Publications, 2015.
- [5] Wampler, D., "**Programação Funcional Para Desenvolvedores Java**". O'Reilly Novatec, 2012.