**Scalable Performance Tuning of Hadoop MapReduce: A OpenSimplex Approach**

Hadoop MapReduce is a popular framework for distributed storage and processing of large datasets and is used for big data analytics. It has various configuration parameters which play an important role in deciding the performance i.e., the execution time of a given big data processing job. Default values of these parameters do not result in good performance and therefore it is important to tune them. However, there is inherent difficulty in tuning the parameters due to two important reasons - first, the parameter search space is large and second, there are cross-parameter interactions. Hence, there is a need for a dimensionality-free method which can automatically tune the configuration parameters by considering the cross-parameter dependencies. In this paper, we propose a novel Hadoop parameter tuning methodology, based on a noisy gradient algorithm known as the simultaneous perturbation stochastic approximation (SPSA). The SPSA algorithm tunes the selected parameters by directly observing the performance of the Hadoop MapReduce system. The approach followed is independent of parameter dimensions and requires only 2 observations per iteration while tuning. We demonstrate the effectiveness of our methodology in achieving good performance on popular Hadoop benchmarks namely Grep, Bigram, Inverted Index, Word Co-occurrence and Terasort. Our method, when tested on a 25 node Hadoop cluster shows 45-66% decrease in execution time of Hadoop jobs on an average, when compared to prior methods. Further, our experiments also indicate that the parameters tuned by our method are resilient to changes in number of cluster nodes, which makes our method suitable to optimize Hadoop when it is provided as a service on the cloud.

|  |  |
| --- | --- |
|  | On the 2nd of October, 2012, NIST decided what algorithm was going to be used to perform hashing. This was the Keccak algorithm.  The Keccak algorithm is based on the [hermetic sponge strategy](http://sponge.noekeon.org/). It's the new standard algorithm. We use standards to make have better compatibility.  Keccak was designed by Guido Bertoni, Joan Daemen (one of the creators of AES), Michaël Peeters, and Gilles Van Assche. They built it based on their [Radiogatùn algorithm](http://en.wikipedia.org/wiki/RadioGat%C3%BAn).  Does this mean SHA-2 is unsafe? No SHA-2 is still considered safe, we just know that in the future it will not be safe anymore and we need assurance that there will be will be an alternative available. We also do not want to transition from one day to another, so therefore they already standardised a new hashing algorithm, so people can have time to change and so that we know we have a secure algorithm at hand when we need it. |