Chapter 1

Introduction

Bio-inspired optimization algorithms are used to find good (preferably best) solutions to a given optimization problem. They mimic the behavior of biological agents. A prominent example is the genetical algorithm (GA) [1] that uses a simplified model of natural evolution to improve the solutions for an optimization problem. Another example is the particle swarm optimization (PSO) [2] that is based on the movement of bird flocks.

Most optimization problems are NP-hard. It is not feasible to explore all possible solutions as this would take to much time. Bio-inspired optimization algorithms can't change this fundamental issue, but in many cases their approximation techniques provide "good enough" solutions to a problem in an acceptable time.

Parallelization techniques can be used to further reduce the computational time or to search through more candidate solutions in the same amount of time. There are different approaches to parallelization on current computer architectures. Some of them focus on the exploitation of processing units that are local to a given machine like SIMD, multi core or specialized hardware (GPU, FPGA, ASIC). This local approaches work well in many cases, but the increase of computational power is limited to the amount of resources that single machine can handle.

Further performance increases can only be achieved by distributing the work to several machines, forming a cluster. The downside of this approach is the increased complexity. On the one side, the machines and their components (e.g. hard drives) are unreliable and may fail. This must be detected and handled in an appropriate way. On the other side, the machines need to communicate to each other through the network to exchange work tasks and results.

Apache Hadoop [3] is an open source software project that provides management tools for clusters and libraries to build distributed applications. It is often referred to as an operating system for clusters. It assumes that cluster hardware is inherently unreliable and provides mechanisms to automatically detect and handle failures. This allows distributed applications to focus on the

implementation rather than on cluster management concerns. Hadoop also provides the mechanisms to start, stop and monitor the distributed applications, automatically restarting them if needed.

Hadoop versions prior to 2.0 were restricted to the MapReduce [4] computational model. This restriction made it difficult to implement e.g. iterative algorithms, like the bio-inspired optimization techniques. The release of Hadoop 2.0 changed the resource management implementation to YARN [5] which makes no assumptions about the executed application.

One drawback of YARN is its lack of support for application specific communication. This restriction comes by design, as YARNs purpose is to manage the cluster, its resources and the running applications.

Different projects try to improve Hadoop and solve the communication issue. Apache Storm [6] for example implements a stream processing model on top of Hadoop, where cluster nodes are connected to form a graph through which the data flows. Data processing and transformation is performed on the nodes. Apache Spark [7] is an implementation that supports the stream model and a batch processing model on top of Hadoop. In addition, it provides a distributed in-memory store [8].

Biohadoop [9] is another project that aims to simplify the implementation of distributed applications on top of Hadoop. It was developed during this thesis and works based on the master - worker pattern. Biohadoop offers an abstract communication mechanism that makes it easy to distribute work items from the master node to any number of worker nodes. Its focus on the master - worker pattern makes it more lightweight than the previous mentioned solutions, but has the drawback to be restricted to the mentioned pattern.

This thesis introduces Biohadoop and demonstrates its usefulness by implementing two bio-inspired optimization techniques on top of it. It provides additional information about Apache Oozie (a Hadoop workflow tool) [10] and how it was extended to support Biohadoop.

The rest of the document is organized as follows: chapter 2 provides an introduction to bio-inspired optimization algorithms as well as an overview of two common representatives: GA and PSO. Chapter 3 delivers information about Hadoop and Oozie that is needed to understand the functionality of Biohadoop. Chapter 4 explains Biohadoops architecture and the implemented modifications for Oozie (section 4.7). Chapter 6 evaluates the performance of Biohadoop using two different implementations of a GA. The conclusions in chapter 7 summarize the master thesis and the obtained results.