

6.1.5 core.reporter

The package `core.reporter` (Figure 7) provides some utilities that can be used to produce a report containing some statistics about the GA execution. The output consists of some CSV (Comma Separated Values) files stored in the HDFS.

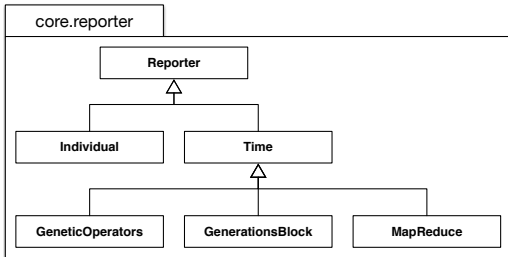


Figure 7: The `reporter` package class diagram.

The `Reporter` class is specialised into two types: `Individual` and `Time`. The former records the chromosome and fitness value of each individual produced during each generation. This allows to keep track and analyse the population evolution. The latter records the execution time (in milliseconds) of: (i) the genetic operators on each node; (ii) the generations; (iii) the MapReduce phases.

All the CSV writing operations are nonblocking so the report functionalities do not influence the computation time of the GA.

6.2 User Level

The `user` package (Figure 8) contains all the base classes the developer should extend in order to implement a specific GA, as explained in Section 5.

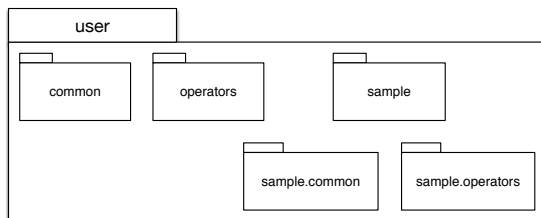


Figure 8: The `user` package class diagram.

In particular, the `common` package contains the `Individual` and `FitnessValue` classes for the definition of the chromosome and fitness value, while the `operators` package contains the classes for the definition of specific genetic operators. Some sample default implementations are included into the `sample` package, such as: sequences of primitive Java types, number fitness values, the random initialisation of sequences, a single point crossover, and the roulette wheel selection.

7. CONCLUSIONS AND FUTURE WORK

This paper described elephant56, a novel framework that allows developers to implement their own GAs in a distributed way on a Hadoop MapReduce cluster of computers.

As future work we plan to empirically evaluate the framework in order to assess its scalability with all the three models (i.e., the global, grid and island model).

Furthermore, it is our intention to enrich the framework with a full documentation for the usage of the framework together with some example problem implementations.

8. REFERENCES

- [1] J. Dean and S. Ghemawat. Mapreduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [2] L. Di Geronimo, F. Ferrucci, A. Murolo, and F. Sarro. A Parallel Genetic Algorithm Based on Hadoop MapReduce for the Automatic Generation of JUnit Test Suites. In *IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 785–793. IEEE, 2012.
- [3] S. Di Martino, F. Ferrucci, V. Maggio, and F. Sarro. Towards Migrating Genetic Algorithms for Test Data Generation to the Cloud. In *Software Testing in the Cloud: Perspectives on an Emerging Discipline*, pages 113–135. IGI Global, 2013.
- [4] P. Fazenda, J. McDermott, and U.-M. O’Reilly. A Library to Run Evolutionary Algorithms in the Cloud using MapReduce. In *European conference on Applications of Evolutionary Computation (EvoApplications)*, pages 416–425. Springer, 2012.
- [5] F. Ferrucci, M.-T. Kechadi, P. Salza, and F. Sarro. A Framework for Genetic Algorithms Based on Hadoop. *arXiv preprint arXiv:1312.0086*, Nov. 2013.
- [6] F. Ferrucci, P. Salza, M.-T. Kechadi, and F. Sarro. A Parallel Genetic Algorithms Framework Based on Hadoop MapReduce. In *ACM/SIGAPP Symposium on Applied Computing (SAC)*, pages 1664–1667. ACM Press, 2015.
- [7] M. Harman. The Current State and Future of Search Based Software Engineering. In *2007 Future of Software Engineering*, pages 342–357. IEEE Computer Society, 2007.
- [8] D.-W. Huang and J. Lin. Scaling Populations of a Genetic Algorithm for Job Shop Scheduling Problems Using MapReduce. In *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 780–785. IEEE, 2010.
- [9] C. Jin, C. Vecchiola, and R. Buyya. MRPGA: An Extension of MapReduce for Parallelizing Genetic Algorithms. In *IEEE International Conference on E-Science (e-Science)*, pages 214–221. IEEE, 2008.
- [10] G. Luque and E. Alba. *Parallel Genetic Algorithms: Theory and Real World Applications*. Number 367 in Studies in Computational Intelligence. Springer, Berlin, 2011.
- [11] A. Verma, X. Llorà, D. E. Goldberg, and R. H. Campbell. Scaling Genetic Algorithms Using MapReduce. In *International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 13–18. IEEE, 2009.
- [12] L. Zheng, Y. Lu, M. Ding, Y. Shen, M. Guoz, and S. Guo. Architecture-Based Performance Evaluation of Genetic Algorithms on Multi/Many-core Systems. In *IEEE International Conference on Computational Science and Engineering (CSE)*, pages 321–334. IEEE, 2011.