

A System Analysis to Dkron Scheduler

Miguel Albuquerque, 1105828¹[1105828]

Instituto Superior Técnico, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
`miguel.albuquerque@tecnico.ulisboa.pt`

Abstract. WRITE ABSTRACT HERE

Repository: `github.com/miguel-msa/dkron-study`

Commit hash: **PUT COMMIT HASH HERE**

Keywords: Dkron · Distributed · Benchmark · Scalability · REVIEW THIS .



1 Introduction

On the surface, Dkron provides an interface that:

1. Has flexible tag-based job definition
2. Allows for Distributed execution, where these jobs can be ran in a distributed mode.
3. Has no Single Point of Failure.
4. Easy to deploy with built-in replicated storage (BuntDB) relying on the Raft protocol.
5. Provides a Web-GUI for administration.

The above characteristics, and more, are further explored in 2

When selecting the system to analyse, some of the factors driving the ultimate choice were:

1. Interest in exploring Go source code.
2. Understand more about a system that is not a database
3. IS THIS OK? REVIEW THIS

After finding some systems fulfilling these requirements, Dkron became the system that raised a curious question: *What is the performance of a scheduler, seemingly, focused on availability? and reliability?*

As this is not explored by Dkron, I became interested in finding out the implementation, e.g. scheduling approach, and extrapolate, joined with benchmarking, its behavior, i.e., its performance, and scaling capacity.

Ultimately, Dkron is a golang written scheduler with specific particularities for some key use-cases. Its claims are not bold, neither on its performance, nor on how it actually does scheduling. This opens an opportunity to not only explore something unclaimed, but also on how Dkron found the optimal point, if exists, between performance and reliability.

The analysis will consider these use-cases and apply a ...benchmark... to analyse the performance and infer Dkron's performance on workloads similar, as in trying to represent, to such use-cases.

2 System Description

PUT HERE ALL THE relevant PERKS&QUIRKS of Dkron in detail

Always available: Using the power of the Raft protocol, Dkron is designed to be always available. If the cluster leader node fails, a follower will replace it, all without human intervention.

Flexible targets: Simple but powerful tag-based target node selection for jobs. Tag node count allows to run jobs in an arbitrary number of nodes in the same group or groups.

2.1 No Single Point of Failure

This no SPOF characteristic is very relevant for systems that depend on workload automation to function i.e. the scheduler dependee system might be fault-tolerant, but the scheduler itself might not, making these "fault-tolerant" systems, indirectly, not as so - Dkron fixes this problem by, as they claim, being the only existing scheduler with no SPOF.

Therefore, with acceptable performance, whilst providing Reliability? and Availability?, Dkron is an interesting solution for use-cases that must guarantee fault-tolerance.

3 Experimental Design

4 Results

5 Conclusion

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	Lecture Notes	14 point, bold
1st-level heading	1 Introduction	12 point, bold
2nd-level heading	2.1 Printing Area	10 point, bold
3rd-level heading	Run-in Heading in Bold. Text follows	10 point, bold
4th-level heading	Lowest Level Heading. Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z$$

(1)

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see

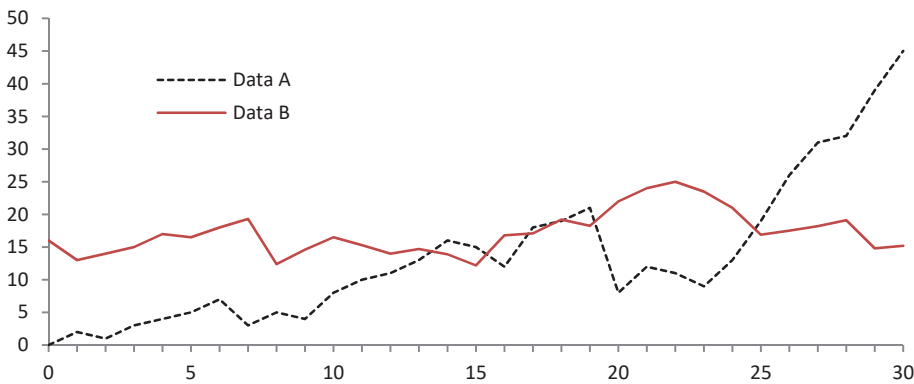


Fig. 1. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], and a homepage [5]. Multiple citations are grouped [1,2,3], [1,3,4,5].

Acknowledgments. A bold run-in heading in small font size at the end of the paper is used for general acknowledgments, for example: This study was funded by X (grant number Y).

Disclosure of Interests. It is now necessary to declare any competing interests or to specifically state that the authors have no competing interests. Please place the statement with a bold run-in heading in small font size beneath the (optional) acknowledgments¹, for example: The authors have no competing interests to declare that are relevant to the content of this article. Or: Author A has received research grants from Company W. Author B has received a speaker honorarium from Company X and owns stock in Company Y. Author C is a member of committee Z.

References

1. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). <https://doi.org/10.1007/1234567890>
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, <http://www.springer.com/lncs>, last accessed 2023/10/25

¹ If EquinOCS, our proceedings submission system, is used, then the disclaimer can be provided directly in the system.