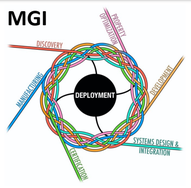
MONGODB ATLAS OPTIMIZES FILE CONTENT MANAGEMENT SYSTEM PERFORMANCE



The U.S. Federal **Materials Genome Initiative (MGI)** exists to accelerate the solving of new materials using informatics, advanced modeling, experimental tools, and quantitative data. The initiative supports development of robust data infrastructures to be queried at scale by materials science, materials engineering, and industrial research communities. The work of this project contributes to this end by exploring a megabyte-sized example of database optimization, a prerequisite for big data-driven materials discoveries.

BACKGROUND

The MGI lends itself to FAIR database models. FAIR (Findable-Accessible-Interoperable-Reusable) dbs are valuable, ease-of-use computational tools that can be applied to materials-based research. A FAIR db must organize metadata and data structures for uniform queries and CRUD ops from thousands of users using various clients. The organizational system used by the Billinge Group for internal dbs is a file content management system called Regolith. The internal dbs at the Billinge Group are growing in size and local file system (fs) backends have HD storage limits. Additionally, the wall times of Regolith tasks are monotonically increasing.

PROJECT SCOPE

Use Case 1

- PI wants to assign projects to students quickly
- PI uses Regolith's projectum adder to assign a new project
- The projectum adder executes instantaneously

Use Case 2

- PI wants to see student progress on project milestones prior to meeting
- PI uses the project lister in Regolith to review milestone statuses
- The project lister provides an instantaneous listing of all milestones
- PI is up to date with milestone progress at start of a meeting

METHODOLOGY

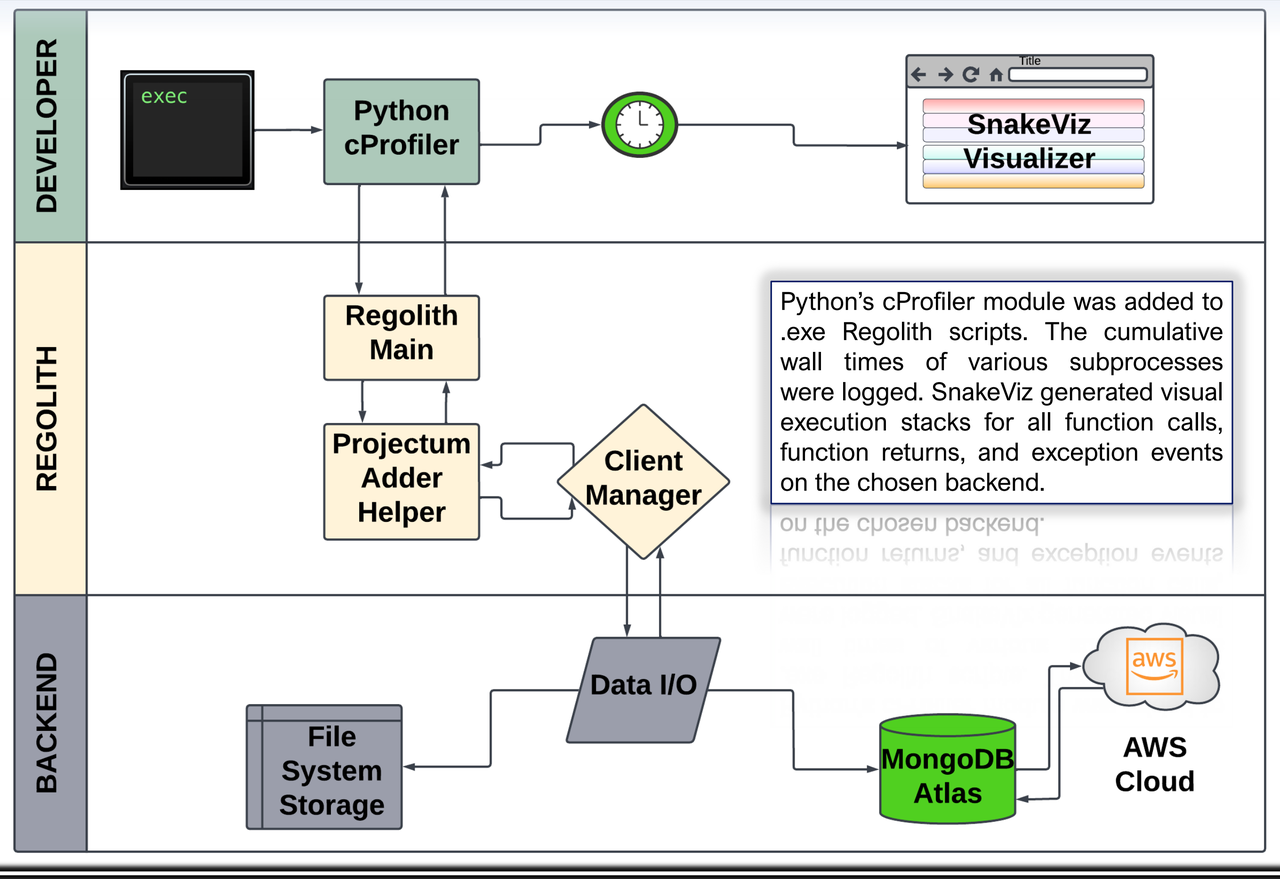


TABLE 1

MONGODB BACKEND IS 3.5 X FASTER !

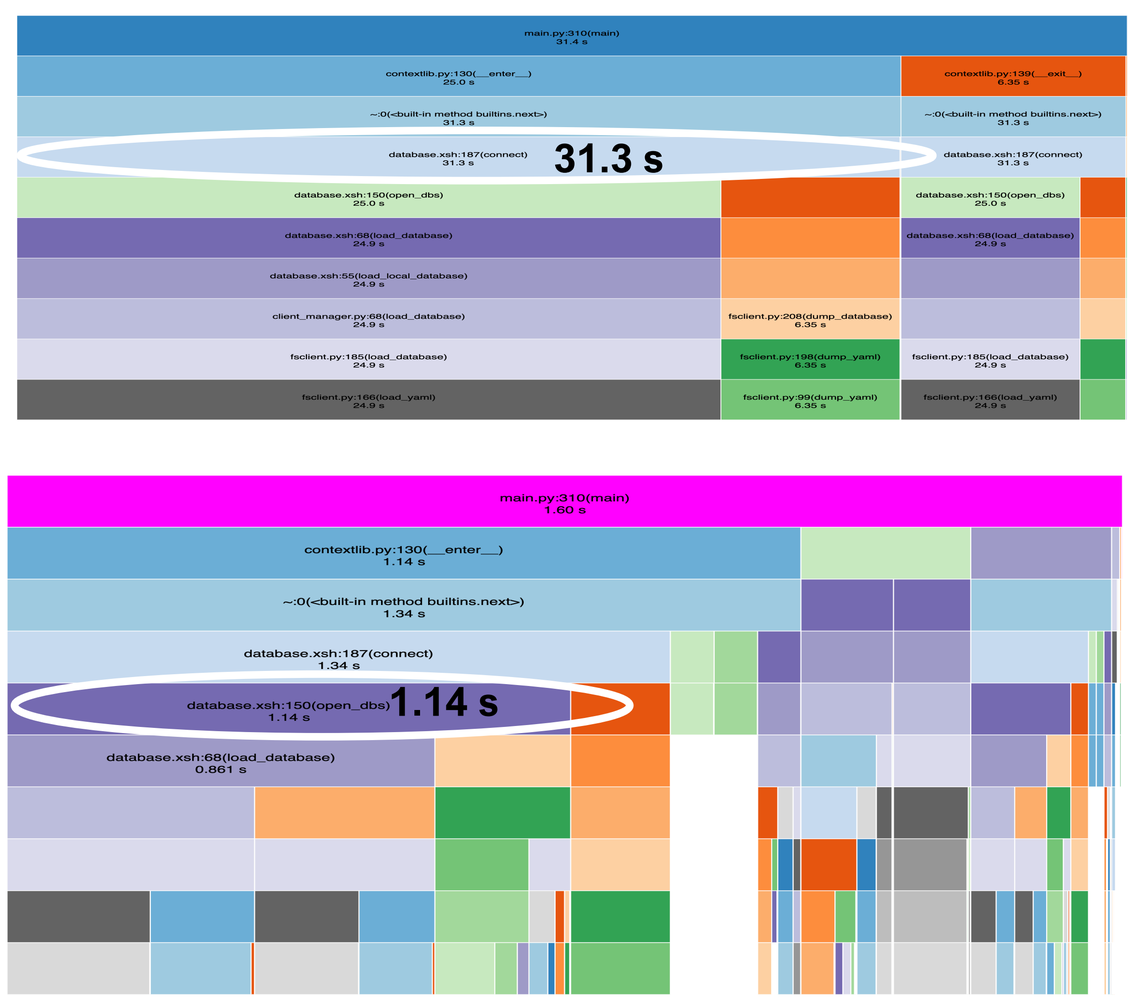
Wall times of Regolith on the MongoDB and fs backends under various conditions for **N trials.**

	No Code Mod (s) (N = 20)	Mods all docs call in projectum adder (N = 20)	No collection load + No database load (N = 20)
MongoDB Atlas Backend	3.606 s	3.390 s	2.939 s
File System Backend	11.832 s	11.654 s	2.348 s

CONCLUSION

- MongoDB Atlas is a faster, more efficient backend than the local fs
- MongoDB is an even better backend as dbs increase in size and scale
- Regolith helper commands did not execute instantaneously in the absence of all db and collection loading calls on the MongoDB backend
- Latency was introduced when initializing the Atlas cluster connection

FIGURE 1



Visual execution stacks of the Regolith projectum adder. Encircled subfunction boxes display the proportional wall times associated with performing the open_dbs function on the **(top)** fs backend and **(bottom)** MongoDB backend.

ACKNOWLEGEMENTS

Many thanks to Prof. Simon Billinge who has so freely shared his incredible talents and wealth of knowledge in materials science with our lab daily. The mentorship I have received from Prof. Billinge, and all members of the Billinge Group, has afforded me invaluable skills in data science, data management, informatics, and project management. After working in this lab, I am thoroughly convinced that a Ph.D in Applied Research and Data Science is my future. I would also like to thank the teams at Alexa AI Columbia Engineering for investing in the SURE Program and for making this experience a defining moment in my undergraduate career.

Dominic Peters, Nick Asker, Simon Billinge, Ph.D.