Sealing pipeline

The process of sealing sectors is called the sealing pipeline. It is important for storage providers to understand the steps of the process.

Each step in the sealing process has different performance considerations, and fine-tuning is required to align the different steps optimally. For example, storage providers that don't understand the process expected throughput may end up overloading the sealing pipeline by trying to seal too many sectors at once or taking on a dataset that is too large for available infrastructure. This can lead to a slowersealing rate , which is discussed in greater detail inSealing Rate .

Overview

The sealing pipeline can be broken into the following steps:

?

AddPiece

The sealing pipeline begins withAddPiece (AP), where the pipeline takes aPiece and prepares it into the sealing scratch space for thePreCommit 1 task (PC1) to take over. In Filecoin, aPiece is data in CAR-file format produced by anIPLD DAG with a correspondingPayloadCID andPieceCID . The maximum Piece size is equal to the sector size, which is either 32 GiB or 64 GiB. If the content is larger than the sector size, it must be split into more than onePieceCID during data preparation.

The AddPiece process is only uses some CPU cores; it doesn't require the use of a GPU. It does write a lot of data on the sealing volume though. Therefore it is recommended to limit the concurrent AP processes to 1 or 2 via the environment variableAP_32G_MAX_CONCURRENT=1 .

It is typically co-located on a server with other worker processes from the sealing pipeline. As PC1 is the next process in the sealing pipeline, running AddPiece on the same server as the PC1 process is a logical architecture configuration.

Consider limiting the AP process to a few cores by using thetaskset command , where is the range on which cores the process needs to run on:

```

Copy taskset-clotus-workerrun...

```

PreCommit 1

PreCommit 1 (PC1) is the most CPU intensive process of the entire sealing pipeline. PC1 is the step in which a sector, regardless of whether it contains data or not, is cryptographically secured. The worker process loads cryptographic parameters from a cache location, which should be stored on enterprise NVMe for latency reduction. These parameters are then used to run Proof-of-Replication (PoRep) SDR encoding against the sector that was put into the sealing scratch space. This task is single-threaded and very CPU intensive, so it requires a CPU with SHA256 extensions. Typical CPUs that meet this requirement include the AMD Epyc Milan/Rome or an Intel Xeon Ice Lake with 32 cores or more.

Using the scratch space, the PC1 task will create 11 layers of the sector. Storage providers must host scratch space for this on enterprise NVMe. This means that:

- Every sector consumes memory equal to 1+11 times its size on the scratch volume.
- For a 32 GiB sector, PC1 requires 384 GiB on the scratch volume
- For a 64 GiB sector, PC1 requires 768 GiB.
-

In order to seal at a decent rate and to make use of all the sealing capacity in a PC1 server, you will maximize the concurrent PC1 jobs on a system. Set thePC1_32G_MAX_CONCURRENT= environment variable for the PC1 worker. You can learn more about this in the chapter onSealing Rate . Sealing several sectors multiplies the requirements on CPU cores, RAM, and scratch space by the number of sectors being sealed in parallel.

The process of sealing a single 32 GiB sector takes roughly3 hours but that time depends largely on your hardware and what other jobs are running on that hardware.

PreCommit 2

When PC1 has completed on a given sector, the entire scratch space for that sector is moved over to thePreCommit 2 (PC2) task. This task is typically executed on a different server than the PC1 server because it behaves differently. In short, PC2 validates PC1 using the Poseidon hashing algorithm over the Merkle Tree DAG that was created in PC1. As mentioned in the previous section, the entire scratch space is either 384 GiB or 768 GiB, depending on the sector size.

Where PC1 is CPU-intensive, PC2 is executed on GPU. This task is also notably shorter in duration than PC1, typically 10 to 20 minutes on a capable GPU. This requires a GPU with at least 10 GiB of memory and 3500+ CUDA cores or shading units, in the case of Nvidia. Storage providers can use slower GPUs, but this may create a bottleneck in the sealing pipeline.

For best performance, compile Lotus with CUDA support instead of OpenCL. For further information, see the Lotus CUDA Setup .

In the case of a Snap Deal , an existing committed capacity sector is filled with data. When this happens, the entire PC1 task does not run again; however, the snapping process employs PC1'sreplica-update andprove-replica-update to add the data to the sector. This can run on the PC2 worker or on a separate worker depending on your sealing pipeline capacity.

When PC2 has completed for a sector, aprecommit message is posted on-chain. If batching is configured, Lotus will batch these messages to avoid sending messages to the chain for every single sector. In addition, there is a configurable timeout interval, after which the message will be sent on-chain. This timeout is set to 24 hours by default. These configuration parameters are found in the.lotusminer/config.toml file.

If you want to force the pre-commit message on-chain for testing purposes, run:

```
Copy lotus-minersectorsbatchingprecommit--publish-now
```

The sealed sector and its 11 layers are kept on the scratch volume until Commit 2 (C2) is complete.

WaitSeed

WaitSeed is not an actual task that is executed, but it is a step in the pipeline in which the blockchain forces the pipeline to wait for 150 epochs as a built-in security mechanism. With Filecoin's 30 second epochs, this means 75 minutes must elapse between PC2 and the next task, Commit 1 (C1).

Commit 1

The Commit 1 (C1) phase is an intermediate phase that performs the preparation necessary to generate a proof. It is CPU-bound and typically completes in seconds. It is recommended that storage providers run this process on the server where C2 is running.

Commit 2

The last and final step in the sealing pipeline is Commit 2 (C2). This step involves the creation of zk-SNARK proof. Like PC2, this task is GPU-bound and is, therefore, best co-located with the PC2 task.

Finally, the proof is committed on-chain in a message. As with the pre-commit messages, the commit messages are batched and held for 24 hours by default before committing on-chain to avoid sending messages for each and every sector. You can again avoid batching by running:

```
Copy lotus-minersectorsbatchingcommit--publish-now
```

Finally, the sealed sector is stored in the miner's long-term storage space, along with unsealed sectors, which are required for retrievals if configured to do so.

Previous Storage provider automation Next Sealing rate

Last updated5 months ago