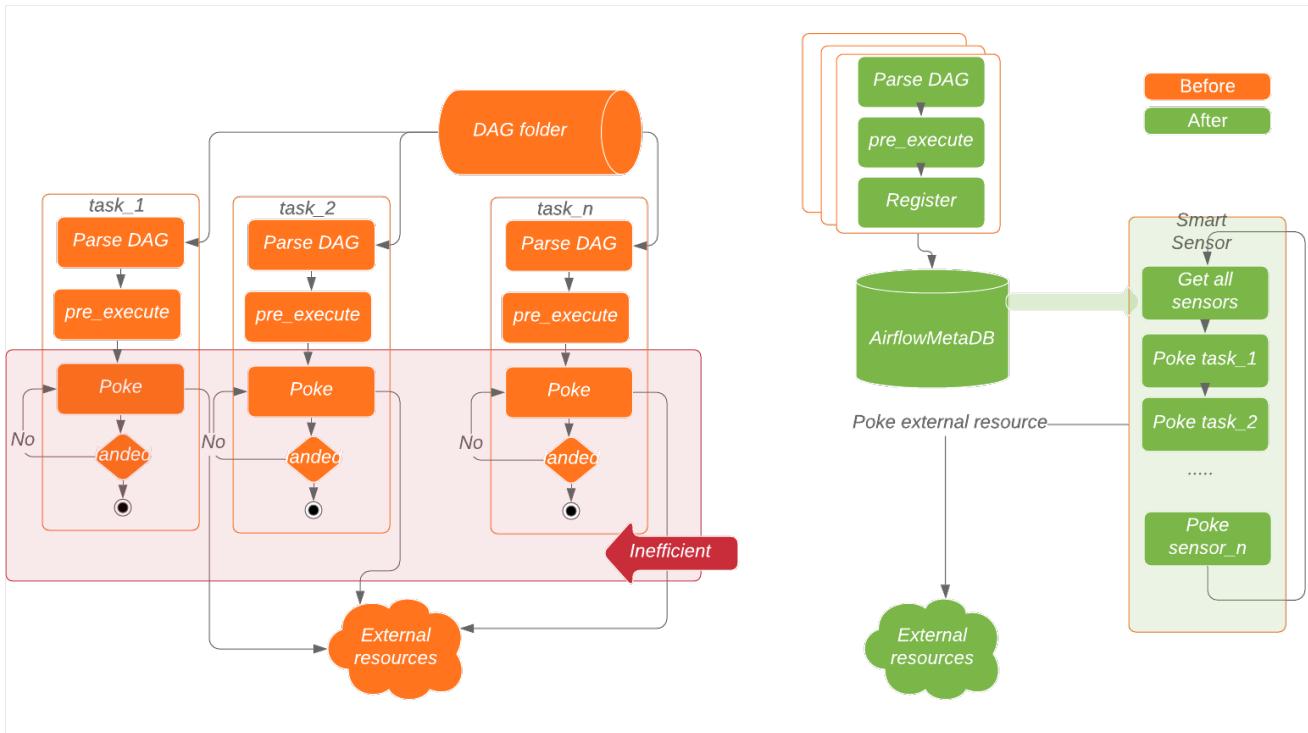


# Smart Sensor

## Warning

This is an **early-access** feature and might change in incompatible ways in future Airflow versions. However this feature can be considered bug-free, and Airbnb has been using this feature in Production since early 2020 and has significantly reduced their costs for heavy use of sensors.

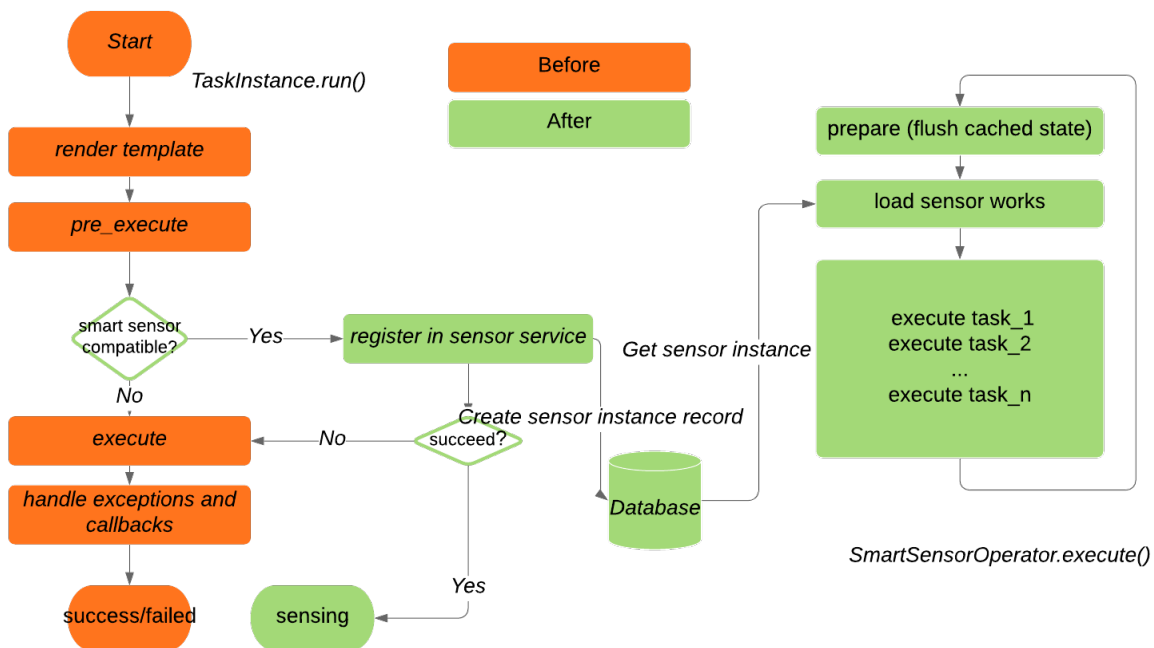
The smart sensor is a service (run by a builtin DAG) which greatly reduces airflow's infrastructure cost by consolidating some of the airflow long running light weight tasks.



Instead of using one process for each task, the main idea of the smart sensor service is to improve the efficiency of these long running tasks by using centralized processes to execute those tasks in batches.

To do that, we need to run a task in two steps, the first step is to serialize the task information into the database; and the second step is to use a few centralized processes to execute the serialized tasks in batches.

In this way, we only need a handful of running processes.



The smart sensor service is supported in a new mode called “smart sensor mode”. In smart sensor mode, instead of holding a long running process for each sensor and poking periodically, a sensor will only store poke context at sensor\_instance table and then exits with a ‘sensing’ state.

When the smart sensor mode is enabled, a special set of builtin smart sensor DAGs (named smart\_sensor\_group\_shard\_xxx) is created by the system; These DAGs contain `SmartSensorOperator` task and manage the smart sensor jobs for the airflow cluster. The SmartSensorOperator task can fetch hundreds of ‘sensing’ instances from sensor\_instance table and poke on behalf of them in batches. Users don’t need to change their existing DAGs.

## Enable/Disable Smart Sensor

Updating from a older version might need a schema change. If there is no `sensor_instance` table in the DB, please make sure to run `airflow db upgrade`

Add the following settings in the `airflow.cfg` :

```
[smart_sensor]
use_smart_sensor = true
shard_code_upper_limit = 10000

# Users can change the following config based on their requirements
shards = 5
sensors_enabled = NamedHivePartitionSensor, MetastorePartitionSensor
```

- `use_smart_sensor` : This config indicates if the smart sensor is enabled.
- `shards` : This config indicates the number of concurrently running smart sensor jobs for the airflow cluster.
- `sensors_enabled` : This config is a list of sensor class names that will use the smart sensor. The users use the same class names (e.g. HivePartitionSensor) in their DAGs and they don’t have the control to use smart sensors or not, unless they exclude their tasks explicitly.

Enabling/disabling the smart sensor service is a system level configuration change. It is transparent to the individual users. Existing DAGs don’t need to be changed for enabling/disabling the smart sensor. Rotating centralized smart sensor tasks will not cause any user’s sensor task failure.

## Support new operators in the smart sensor service

- Define `poke_context_fields` as class attribute in the sensor. `poke_context_fields` include all key names used for initializing a sensor object.
- In `airflow.cfg`, add the new operator’s classname to `[smart_sensor] sensors_enabled`. All supported sensors’ classname should be comma separated.

Was this entry helpful?



Want to be a part of Apache Airflow? [Join community](#)

License Donate Thanks  
Security  
© The Apache Software Foundation 2019

Apache Airflow, Apache, Airflow, the Airflow logo, and the Apache feather logo are either registered trademarks or trademarks of The Apache Software Foundation. All other products or name brands are trademarks of their respective holders, including The Apache Software Foundation.