Update on conditions DB deployment

Lino Gerlach, Paul Laycock, Ruslan Mashinistov

14.12.2022

@BrookhavenLab

Overview

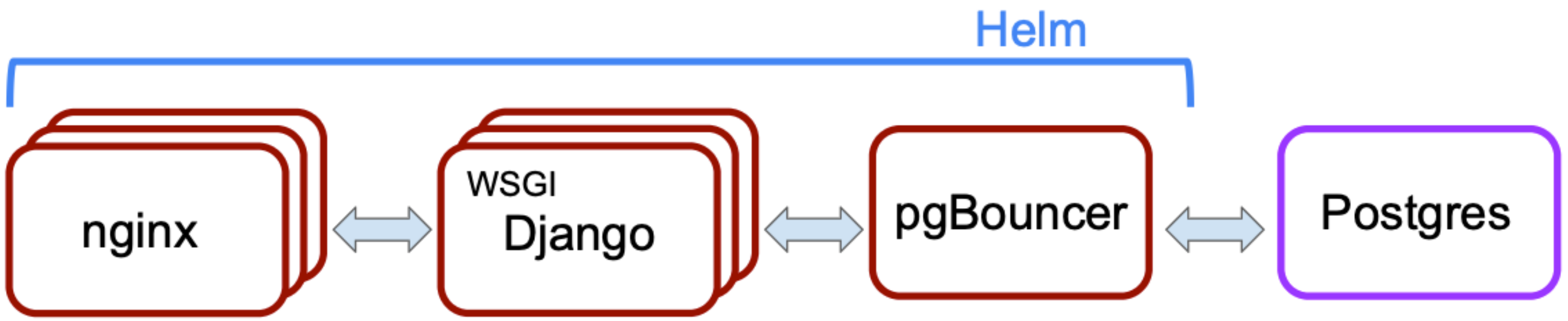
Long term goal:

- Develop conditions DB for DUNE
  - ProtoDUNE as ‘testing ground’
- Use existing, experiment-unspecific database ‘nopayloaddb’
  - Designed according to HSF recommendations
- Deploy on Fermilab resources

In today’s talk:

- Deployment configuration
- State of C++ client-side library
- Instance at BNL for sPhenix
- Plan for deployment at Fermilab

Deployment of nopayloaddb via OKD



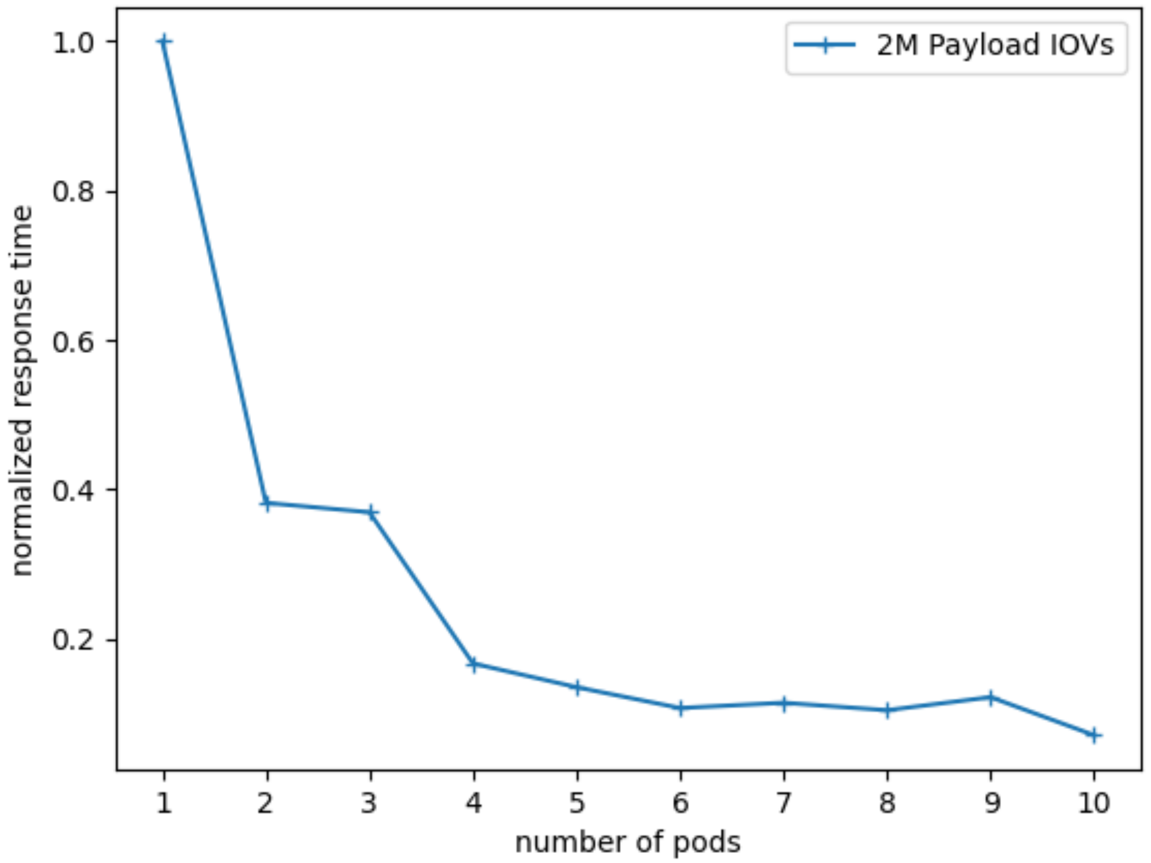
- Single command-line deployment
- Helm configuration (values) defines main parameters: OKD (Open Shift) project name, db credentials, URL …

From Ruslan Mashinistov

- Each ’Service’ consists of several ‘Pods’ (can be scaled)
- Postgres backend on some persistent storage
- Payloads are stored on a POSIX file system

Scaling w/ number of pods

- Vary number of nginx and django pods simultaneously (n each)
- Repeat full test campaign for each setup
    - 100 HTC jobs making 100 calls each



- Average response time decreases with number of pods
- Flattening at ~5 pods per service
- Can be easily scaled up if needed

Client-side tool - Functionality

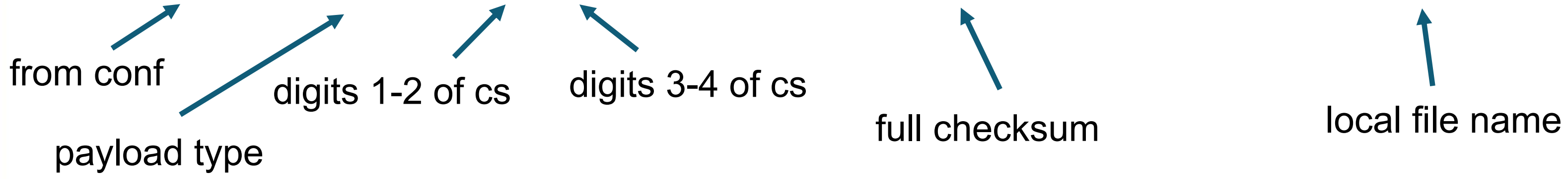
Stand-alone C++ tool to communicate with nopayloaddb

- Experiment unspecific
  - (Proto)DUNE specific stuff only in LArSoft Service
- Read & write operations on DB
  - Basically wrapper for libcurl
- Local caching
- Handling of payload files
  - Keep DB and payload store synchronized

Client-side tool – Payload Handling

- Writing all payload files with user-defined names to single dir not an option
- Use md5 checksum to generate remote file name. Example:

/prefix/example_pt/2c/1c/2c1c9467a370028344ac486d438c2ae2_file.dat



<u>Pro's</u>
- Avoids duplication of payload files
- Results in evenly distributed payload files across 16^2 * 16^2 ≈ 65k dirs.
  - Good for performance (optimal: 100-1000 files & subdirs per dir)

<u>Con's</u>
- Cannot easily browse payload files by hand

nopayloaddb @ sPhenix

- sPhenix: new heavy-ion detector @BNL
  - Aims to start taking data end of March 2023
- Developed experiment-specific version of client tool
  - Minor modification w.r.t. experiment-unspecific client
- Using nopayloaddb as backend for several months now
- Write payloads to local file system
  - ‘publish’ payloads to /cvmfs/ hourly via cron job
- Read from local fs if not published to /cvmfs/ yet
- Started using client tool two weeks ago
  - Provide useful feedback from real-world use as conditions DB

Questions for deployment at Fermilab

- Possible problems with different software versions at BNL vs Fermilab?
  - @BNL: Kubernetes: v.1.23.5, OKD: 4.5.0-0, Helm: v3.4.1
- What are the available File systems?
  - Where to store payloads?
  - Where to write backups and logs?
- Existing postgres backend?
  - @BNL: postgres backend as a pod in project

Time schedule

- Rest of the year:
  - Keep collecting feedback from sPhenix users
    - Possibly implement changes to client tool & nopayloaddb
  - Conduct some additional scaling tests

- Early next year (January):
  - Have a meeting with Fermilab experts regarding open questions
  - Deploy on Fermilab resources