- stripJobParams in shared_file_messenger to strip redudant attributes from jobSpec like prodDBlock before sending it to Yoda
  - 13MB of a jumbo JobSpec → 460kB
  - To avoid MPI message deadlock between Yoda and Droids
- Two issues for Yoda+Jumbo+HPC commissioning: Slow completion of ES tasks and more intuitive ES monitoring
  - Working with Nicolo and Wen
  - Other issues to be addressed later

Slow Completion of ES tasks

- Task parameters to define job attempts
  - maxAttempt : The max number of reattempts for each input. Typically 30, defined by prod managers
  - maxFailure : The max number of failed reattempts for each input. Typically 10, defined by prod managers
  - maxAttemptES : The max number of reattempts for each event. 10 by default
  - maxAttemptEsJob : The max number of reattempts for each ES consumer. 10 by default
  - nEsConsumer : The number of ES consumers concurrently running for each input. 5 by default
- Input = a file, an event set in a file, or a group of files, depending on the task definition
- In the worst case scenario, the task goes to failed after (maxAttempt x maxAttemptEsJob x nEsConsumer) jobs are generated → Too many attempts → Long tail caused by tasks with problematic inputs/events
- The default value for maxAttemptES and maxAttemptEsJob is not very small since
  - events were discarded every time the input was retried, and
  - every time events are sent to the pilot that is regarded as one attempt even if they are not processed, which typically happens the pilot prefetches too many events for some special resources
- New task params
  - notDiscardEvents : to keep events when inputs are retried
  - decAttOnFailedES : to change attempt numbers of events only when they are failed
- New default values
  - maxAttemptES:1, maxAttemptEsJob:1, notDiscardEvents:True, decAttOnFailedES:True
  - Wen is trying a couple of test tasks. Will use them by default if test tasks are fine
  - Would be better to set maxAttempts = max(maxFailure, maxAttempt/nEsConsumer) too

Input-based Monitoring for ES

- Current task page is job-centric, which doesn't fit well for ES workflow
  - Many consumers + merge jobs work for a single input
  - A consumer is a job in terms of DB record and processes only a part of input
    - No 1-to-1 mapping between job and input
  - Constructing task pages by resolving relationship among consumers and merge jobs
  - Status of each consumer is not so important
    - E.g. it is OK even if one consumer is stuck as long as another consumer is running to process the input
- A new task page with summary of input status would be more useful
  - Possible input status
    - queued : consumers are submitted for the input to Panda, but they are not running, i.e. they are in assigned/activated/starting
    - running : one or more consumers are running for the input
    - merging : being merged by a merge job
    - transferring : output files are being transferred to the destination
    - finished : the input was successfully processed
    - failed : reaches max attempt
    - ready : the input is active but there are no consumers or merge jobs
  - Information can be obtained from two tables, JEDI_Dataset_Contents and filesTable4
    - ready, finished, and failed only from JEDI_Dataset_Contents
    - others from two tables + lookup to job tables
      - For each input, take PandaIDs with the current attempt from filesTable4, get jobStatus and eventService from job tables, and then e.g. merging if there is an active job with eventService=2 (merge job). This may require advices from DBA to speed up, or new column in JEDI_Dataset_Contents updated by panda/JEDI
  - Jumbo jobs are tricky since they don't have input records in the DB
    - The number of events in each status would be enough since each jumbo works for the entire task
    - sum(actualCoreCount) of running jobs may be useful