Basic Idea

The purpose of the described system it to enable users to run bumps fitting algorithms simultanously on a multiprocessors system.

The purpose of the system described in this document, is to enable multiple users run bumps on multi-processing

Fitting job may take a few hours to complete. One way of shortenning prcessing time is to use multi processing system.

During that time, the system will send sttaus updates to the sending user.

Users will submit fitting job request from client machine to the system server.

A queue manager, running on the server, first verifies correct request structures. If valid, it will add the new job to the end of a First In First Out (FIFO) queue. That queue will be implemented in a relational database (RLDB).

A Queue Interface, also running on the server, will send jobs, from the top, of the queue to a distributed tasks queue system, such as Slurm or Celery. When the job is completed, the queue manager will save the result in the database and delete them from the disk, remove the job from the Slurm/Celery and add the next job.

During the job run time, status updates are being sent to the user.

When the job is completed, the user gets notification, and the results are being stored in the database, as BLOB.

The system components and interconnection are described in the following figure.



From the queue, jobs are sent to will be performed

, that is implemented in a database table. A job in the table can be idle, running.or completed. The server sends the first idle job from the database to a distributed tasks queue, such as Slurm or Celery.

When the job is comleted, the status

The first job

will be . , running a queue. That queue We'd like the user to focus on the fitting itself, and not worry about harware and communication. Also, we'd like to user to get updates on the