General Solve one password after another, let master calculate permutations, chunk them and let the workers do the hashing

Worker *in*: Receives list of permutations P and target hashes H *does*: calculate hashes of permutations (H‘)
returns: subset of given hashes H that occur in H‘ (we will refer to it later on as *result*)

1. Worker registers at Master: Whenever a worker registers, a task is assigned (if no task is left, the next password gets solved, if no password is left, the next chunk is requested; onChunkReceived parses PW, stores them in queue & assigns each worker a task)
2. The master calculates the permutations chunk-wise using the PermutationsGenerator (that uses an iterative approach for calculating the permutations)
3. When a worker has messaged the master a result, it’s content is checked and the job gets updated, the master assignes the worker the next task from the same password cracking job (or the next password if there is no task for the same job left)
4. As soon as all hints have been solved, the password cracking starts

Robustness

When a termination messages is received by Master, the sent messages (which are buffered until a result for them are received) get forwarded to all other workers.

Task

- permutations
- targetHashes
- passwordJobId

Result

- resolvedHashes (hash -> plain)
- passwordJobId