Appendix: API of parallel patterns

Task Queue (Queue.h):

1. ***tq newtq();*** : Create a new task queue.
   Return: An empty task queue.
2. ***void puttask(tq tq, void * v);*** : Put a parameter into a queue
   tq: The task queue you want to put in the parameter.
   v: The value of the parameter
3. ***void *gettask(tq tq);*** *:* Read a value from the queue.
   tq: The queue to be read.
   Return: The value read from the queue.

Farm (parpat.h):

1. ***void *createfarm(void *(*Worker)(void *i), int n, void *p, int maxthread);*** : Create a farm whose workers use the same parameter.
   Worker: Worker function.
   n: Number of workers.
   p: The parameter used by all workers.
   maxthread: The maximum number of threads.
   Return: A task queue of output.
2. ***void *createfarm_queue(void *(*Worker)(void *i), tq tq1, int maxthread);*** : Create a farm with parameters passed by a task queue..
   Worker: Worker function.
   tq1: The task queue for passing parameters.
   maxthread: The maximum number of threads.
   Return: A task queue of output.
3. ***void *createfarm_array(void *(*Worker)(void *i), int n, void *buf[], int maxthread);*** : Create a farm with parameters passed by an array..
   Worker: Worker function.
   n: Number of workers.
   buf: The array for passing parameters.
   maxthread: The maximum number of threads.
   Return: A task queue of output.

Worker Queue (Queue.h):

Relation between worker queues and pipeline patterns is indicated in the report.

1. ***wq newwq(void *(*Worker)(void *i), int maxthread, void *p, int count)****; :* Create a worker queue for pipelines whose workers of the first stage use the same parameter.
   Worker: Worker function of the first stage;
   maxthread: The maximum number of threads in the first stage.
   p: The parameter used by all workers in the first stage.
   count: Number of workers in each sage of the pipeline
   Return: A worker queue with one node in it.
2. ***wq newwq_queue(void *(*Worker)(void *i), int maxthread, tq tq1)****; :* Create a worker

queue for pipelines conveying data by task queues between stages.

Worker: Worker function of the first stage;

maxthread: The maximum number of threads in the first stage.

Tq1: The task queue for passing parameters for the first stage.

Return: A worker queue with one node in it.

3. ***wq newwq_array(void *(*Worker)(void *i), int maxthread, int count)****; :* Create a worker queue for pipelines conveying data by arrays between stages.

   Worker: Worker function of the first stage;

   maxthread: The maximum number of threads in the first stage.

   count: Number of workers in each sage of the pipeline.

   Return: A worker queue with one node in it.

4. ***void putworker(wq wq, void *(*Worker)(void *i), int maxthread); :***Add a node to a worker queue, which also means add a stage to the pipeline.

   wq: The worker queue where you want to add a node.

   Worker: Worker function of added stage;

   maxthread: The maximum number of threads in added stage.

Pipeline (parpat.h):

5. ***void *createpipe(WorkerQueue wq);:*** Create a pipeline pattern whose workers in the first stage use the same parameter.

   wq: The worker queue with information of stages of the pipeline.

   Return: A task queue of output of the final stage.

6. ***void *createpipe_queue(WorkerQueue wq);:*** Create a pipeline pattern conveying data by task queues between stages.

   wq: The worker queue with information of stages of the pipeline.

   Return: A task queue of output of the final stage.

7. ***void *createpipe_array(WorkerQueue wq, void *buf[]);:*** Create a pipeline pattern conveying data by arrays between stages.

   wq: The worker queue with information of stages of the pipeline.

   buf: The input array of the first stage.

   Return: A task queue of output of the final stage.

Return Value (parpat.h)*:*

1. ***void send_result(void * r);***: Return a value.

   r: The returned value.