

ComfXYZ:任务调度系统

向AI提问的Prompt

你好，简单讲解一下我们的系统：我们系统的技术栈是NextJS+tRPC的全栈框架，搭配了MySQL、Redis、Minio的数据库。

目前在任务调度问题上卡住了，不知道怎么解决这个问题。，我希望您阅读我们的需求以后，开始解决问题。

任务

我们系统中有多个「大任务」，大任务中又有许许多多的「子任务」，还有服务器表，表中有多个「GPU实例」

这些子任务是图像生成任务，他会占用我们的GPU实例长达20秒钟，并且一个GPU实例同一时刻只能运行一个图像生成任务。

我们现在的做法是，当用户点击「创建任务」的时候，则根据参数在数据库中插入一个大任务和若干个子任务。然后就没有然后了。

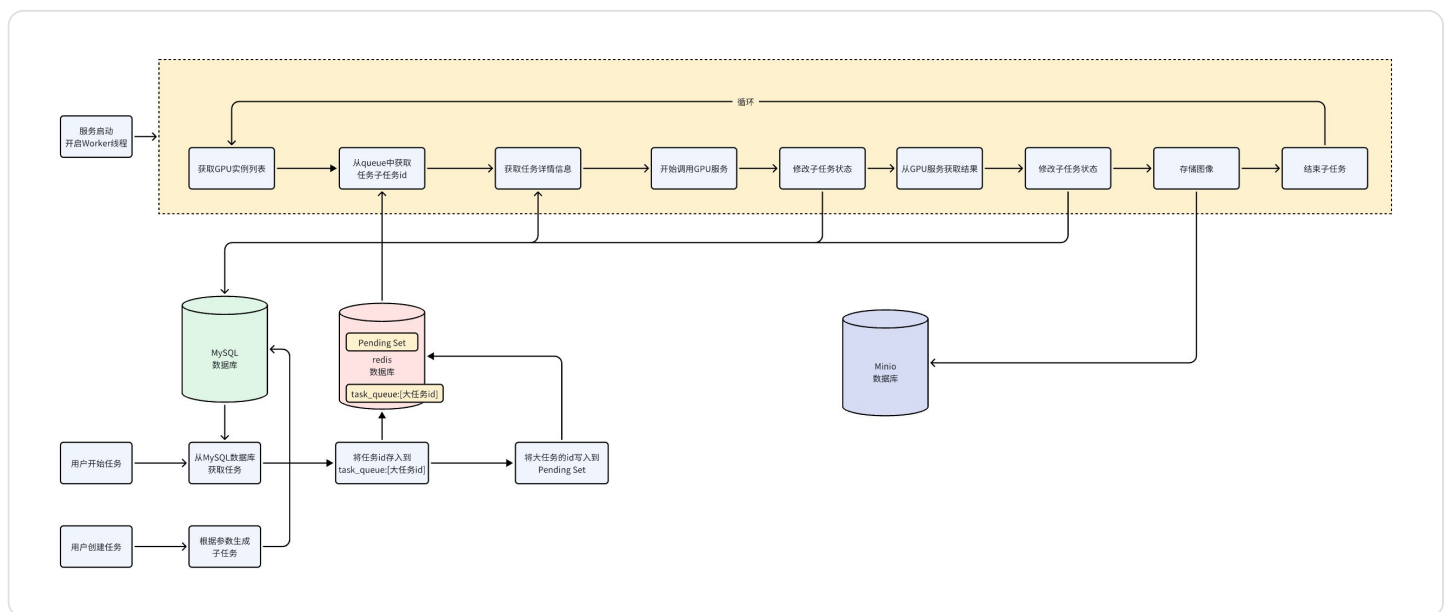
```
1  enum TaskStatus {
2      INIT
3      PENDING
4      RUNNING
5      COMPLETED
6      FAILED
7      CANCELLED
8  }
9
10 model Task {
11     id          Int          @id @default(autoincrement())
12     createdAt   DateTime     @default(now())
13     updatedAt   DateTime     @updatedAt
14     status      TaskStatus   @default(PENDING)
15     workflow    Workflow     @relation(fields: [workflowId], references: [id])
16     workflowId  Int
17     owner       User          @relation(fields: [ownerId], references: [id])
18     ownerId     Int
19     name        String?
20     items       TaskItem[]
21 }
22
```

```

23  model TaskItem {
24      id          Int          @id @default(autoincrement())
25      createdAt   DateTime     @default(now())
26      updatedAt   DateTime     @updatedAt
27      status      TaskStatus   @default(PENDING)
28      task        Task          @relation(fields: [taskId], references: [id])
29      taskId      Int
30      params      Json          // 完整的参数配置
31      result      Json?         // 执行结果
32      error       String?
33      xValue      Json?
34      yValue      Json?
35      zValue      Json?
36  }

```

我的想法是，当用redis实现一个队列，当用户点击了「开始任务」，我们将任务的id放到redis的一个key中



任务系统中有三条主线：用户开始任务、用户创建任务、服务启动开启worker线程

1. 用户创建任务主线

- 创建子任务：用户创建任务后，系统根据参数生成子任务。这是任务产生的源头，为后续处理提供具体工作单元。
- 任务入队相关操作：生成的任务信息会与数据库交互，将大任务 ID 存入 Redis 数据库的 `task_queue`（大任务 id），同时将大任务的 id 写入 Pending Set。这一步是为了让任务进入待处理队列，便于后续服务获取和处理。

2. 用户开始任务主线

- 从 MySQL 获取任务：用户开始任务时，系统从 MySQL 数据库获取任务。这意味着用户启动任务执行流程，系统从存储任务信息的 MySQL 中读取相关数据。

- 后续关联：获取任务后，后续流程与其他主线汇聚，进入从 Redis 的 `task_queue` 获取任务子任务 id 等后续处理步骤，即开始调用 GPU 服务、修改子任务状态、获取结果、存储图像等操作。

3. 服务启动开启 Worker 线程主线

- 启动准备：服务启动并开启 Worker 线程，这是整个任务处理的执行环境准备工作。Worker 线程将负责具体的任务执行调度。
- 获取 GPU 实例列表：Worker 线程获取 GPU 实例列表，为后续调用 GPU 服务做准备，知晓当前可用的 GPU 资源情况。
- 后续任务处理循环：接着从 `task_queue` 获取任务子任务 id，然后依次进行获取任务详情信息、调用 GPU 服务、多次修改子任务状态、获取结果、存储图像等操作，完成一个子任务处理后，若有剩余任务则继续循环处理。

这三条主线分别从任务产生、任务启动执行以及服务执行环境准备三个角度出发，最终在任务处理环节相互关联，共同构成一个完整的任务处理流程。

现在的问题是，nextjs怎么实现「服务启动开启 Worker 线程主线」？你有什么好的想法吗？