

## **Travail par Lots ou Un par Un**

Aujourd’hui, dans notre appartement, l’approvisionnement en eau a rencontré quelques problèmes et n’a pas pu fournir d’eau pendant un certain temps. Après avoir terminé le repas et en regardant les piles de vaisselle que mes proches et moi avions laissées, plusieurs questions m’ont traversé l’esprit.

L’une des questions est de savoir comment faire fonctionner le lave-vaisselle sans alimentation en eau. Il pourrait être conçu pour être connecté à un seau d’eau. De plus, l’embout d’attache du tuyau d’eau devrait être flexible pour permettre un passage facile entre l’alimentation en eau publique et un seau d’eau privé, fabriqué soi-même.

Un autre point à considérer est la question de savoir s’il faut effectuer la tâche par lots ou une par une. Nous pouvons laver la vaisselle après chaque repas ou attendre de la laver après une journée ou quelques jours. Cela dépend de notre manière de prendre les repas et de laver la vaisselle. Nous pouvons également aborder ce problème en tenant compte de la capacité d’un lave-vaisselle.

Cela me rappelle la programmation. Nous pouvons effectuer des tâches par lots ou une par une.

Faire le travail par lots entraîne un problème apparent : il nécessite plus de ressources. Il faut plus de vaisselle car nous retardons le lavage, et il faut plus d’espace mémoire car nous accumulons les données pour différer leur traitement.

Dans la vie réelle, il existe une limite à la quantité d’espace ou au nombre d’objets que l’on peut gérer en même temps. Par exemple, le lave-vaisselle peut probablement contenir au maximum une vingtaine d’assiettes, un peu comme un programme a une limite de mémoire sur un ordinateur ou comme une route a une limite sur le nombre de voitures qui peuvent y circuler.

Il y a aussi le problème de savoir comment séparer la tâche. Devrions-nous la séparer un élément à la fois ou trois éléments à la fois ?

Pour les plats ou les voitures, il est simple de traiter chaque élément comme une unité. Cela signifie qu’un plat est un plat, et une voiture est une voiture. Normalement, ils ne peuvent pas être décomposés en plus petits éléments. Bien qu’il existe encore des exceptions, comme un gros camion qui transporte plusieurs voitures ; un gros camion peut être décomposé en une grande unité et plusieurs voitures traversant la route.

En programmation, c’est beaucoup plus flexible. Même une insertion ou une mise à jour SQL peut être décomposée en plus petits morceaux, sans parler d’un travail de téléchargement,

d'une recherche DFS ou d'une requête.

OK, maintenant que nous avons réfléchi à l'unité de traitement, la question suivante est de savoir combien d'unités nous devrions traiter en un seul lot. Cela peut être n'importe quel nombre entre un et le nombre total d'unités.

La question ici est de savoir si le nombre de lots pour un travail peut être fixe ou dynamique. Pour l'IA générative, le nombre total de caractères du texte d'entrée est flexible. Il existe certaines limites de contexte ou d'entrée, mais dans sa plage de limites, il est flexible.

Lorsque vous utilisez un lave-vaisselle, son espace intérieur a une limite. Dans cette limite, le nombre de vaisselles est flexible. Nous mettons normalement autant de vaisselles que nécessaire dans la machine.

Pour les programmes, le lot de requêtes SQL que la base de données peut traiter en une seule fois a une limite. Dans cette limite, le nombre de requêtes SQL qu'elle peut gérer est flexible. Cependant, nous devons prendre en compte la tâche réseau consistant à transmettre les requêtes SQL du client au serveur de base de données, le temps que l'utilisateur peut attendre, et ce qui se passe si une tâche unitaire du lot échoue.

Ainsi, pour le problème du nombre de tâches unitaires que nous devrions effectuer en un seul lot, nous devons prendre en compte l'objectif du travail, la limite du consommateur ou du gestionnaire en aval, ainsi que la probabilité d'échec.

Cette manière de penser peut s'appliquer à de nombreuses choses. Il y a essentiellement deux problèmes à considérer : quelle est la tâche unitaire et combien d'unités devrions-nous traiter par lot. En considérant ces problèmes, nous pouvons arriver à une solution optimale.