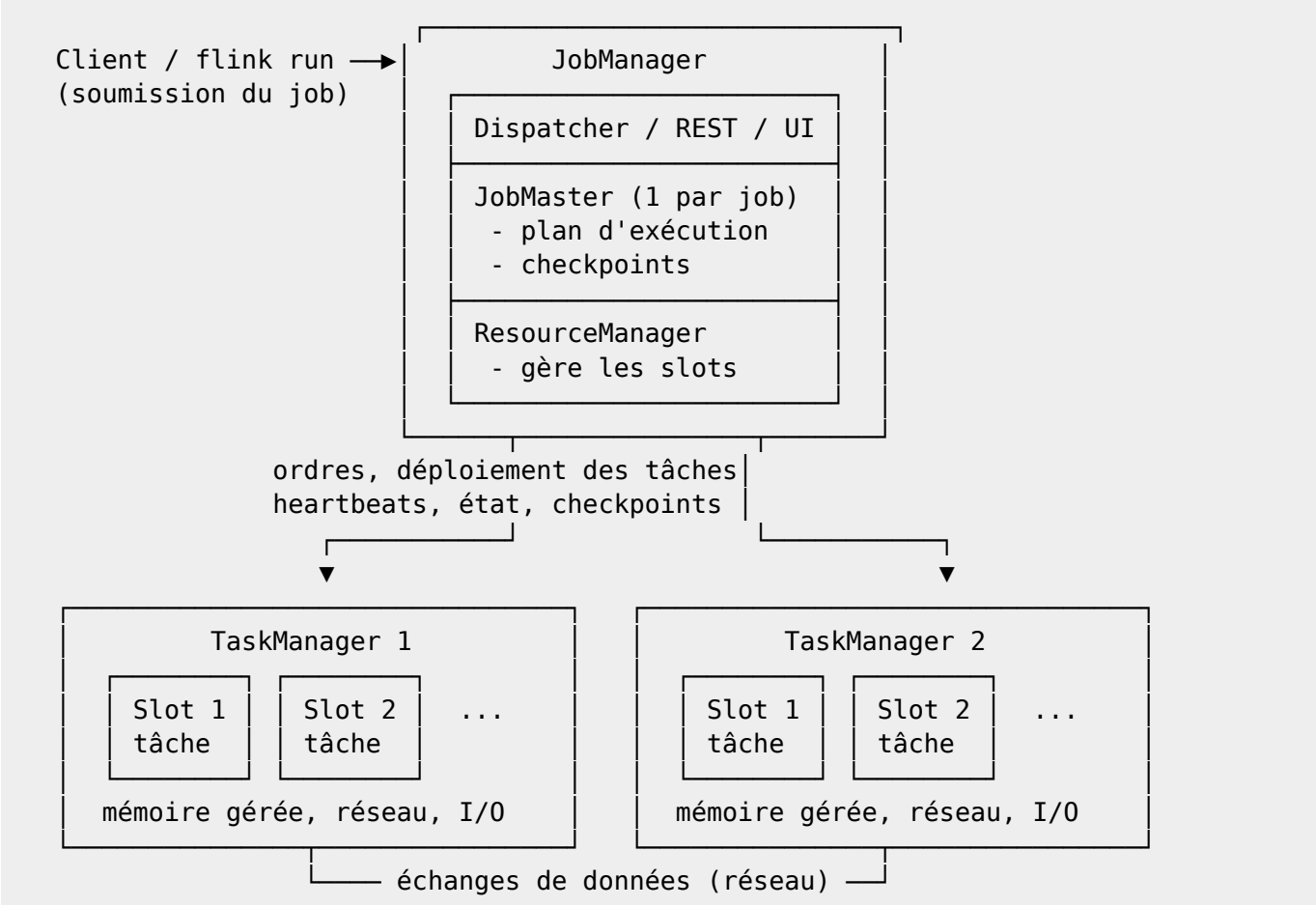
JobManager, TaskManager et slots dans Flink

Vue d'ensemble



Rôles

Composant	Rôle
JobManager	Chef d'orchestre : reçoit les jobs, planifie l'exécution, déploie les tâches, déclenche les checkpoints, gère les pannes.
TaskManager	Processus (JVM) worker qui exécute les tâches et gère mémoire, réseau et I/O.
Slot	Unité de ressources d'un TaskManager (part fixe de sa mémoire gérée). Exécute une sous-tâche ou une chaîne de sous-tâches.

Relations

Relation	Description
JobManager → TaskManager	Le JobManager pilote : il déploie les tâches, surveille et déclenche les checkpoints.
TaskManager → JobManager	Le TaskManager s'enregistre, annonce ses slots, envoie des heartbeats et le statut des tâches.
TaskManager ⊃ slots	Un TaskManager contient N slots (relation 1 à N). Chaque TaskManager a ses propres slots, non partagés.

Relation	Description
Slot ← sous-tâche	Un slot exécute une sous-tâche (ou une chaîne de sous-tâches) du job.
TaskManager ↔ TaskManager	Échange direct des données entre sous-tâches (redistribution au keyBy).

Calcul des slots

```
Total slots = nb de TaskManagers × slots par TaskManager
```

- Le **cluster offre** N × S slots (configuration : `taskmanager.numberOfTaskSlots`, nombre de conteneurs TaskManager).
- Le **job demande** autant de slots que son parallélisme maximal (avec le slot sharing).
- Condition : `parallélisme ≤ N × S`, sinon le job reste en attente de ressources.

Exemple : parallélisme et slots

1 TaskManager de 4 slots, job de parallélisme 2 :

```
TaskManager (4 slots)
┌──────────┬──────────┬──────────┬──────────┐
│ Slot 1   │ Slot 2   │ Slot 3   │ Slot 4   │
│ source→  │ source→  │  libre   │  libre   │
│ map→     │ map→     │          │          │
│ reduce   │ reduce   │          │          │
│ (inst.1) │ (inst.2) │          │          │
└──────────┴──────────┴──────────┴──────────┘
```

Cluster	Parallélisme du job	Résultat
1 TaskManager × 4 slots	2	2 slots utilisés, 2 libres
1 TaskManager × 4 slots	4	Tous les slots utilisés
1 TaskManager × 4 slots	6	Pas assez de slots : le job attend
2 TaskManagers × 2 slots	4	Tous les slots utilisés, répartis sur les 2 TaskManagers

Lien avec Kafka

- Le parallélisme de la **source Kafka** est borné par le nombre de partitions du topic (3 dans le TD).
- Le parallélisme **total du job** est borné par le nombre de slots (4).
- Au-delà de 3, certaines sous-tâches de la source restent inactives.

À retenir

- Le **parallélisme** détermine le nombre de slots *utilisés*, pas le nombre de TaskManagers.
- Si un TaskManager tombe, tous ses slots disparaissent : le JobManager replace les tâches et restaure l'état depuis le dernier checkpoint.
- Dans le TD (mode session, 1 TaskManager), aucun échange réseau entre machines : tout est sur la même JVM.

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/eadl/bloc5/fm4/chap1-b?rev=1795082781

Last update: **2026/11/19 11:06**