

Comparaison de Systèmes Distribués : RMI vs Agents Mobiles

Mano Dinnat Creusier

Alexis Dieu

13 janvier 2026

1 Objectif du projet

L'objectif de ce projet est de comparer deux types de systèmes distribués : RMI et les agents mobiles. Pour cela nous avons implémenté deux scénarios et pour chacun d'entre eux nous avons mesuré le temps d'exécution.

2 Scénario 1 : Somme de valeurs dans un fichier CSV

Dans ce scénario, le traitement porte sur un fichier CSV présent sur un serveur. Pour chaque ligne du fichier, on récupère la valeur de la colonne "nombre" puis on l'ajoute à une somme. Le serveur ne permet la récupération que d'une ligne à la fois. En utilisant RMI, on est donc obligé de faire un appel par ligne. Avec l'agent mobile, un agent est envoyé sur le serveur et réalise le même nombre d'opérations localement avant de revenir avec le résultat seulement.

Résultats du Scénario 1

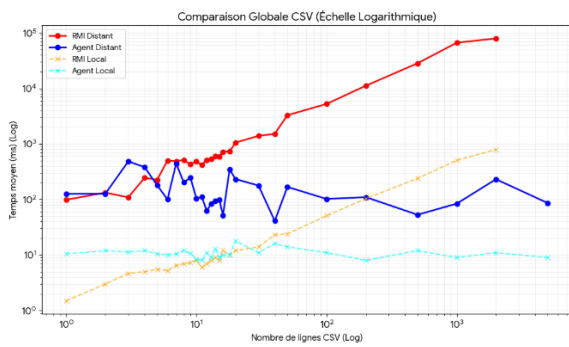


FIGURE 1 – Scénario CSV – Vue Logarithmique

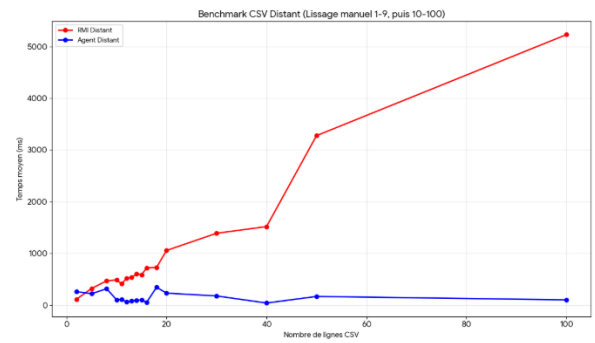


FIGURE 2 – Scénario CSV

TABLE 1 – Scénario CSV (Distant)

N	RMI (ms)	Agent (ms)	Vainqueur	Facteur
1	~95	~126	RMI	$\times 1.3$
10	~486	~105	Agent	$\times 4.6$
100	5 228	~102	Agent	$\times 51$
1 000	66 151	~84	Agent	$\times 787$

3 Scénario 2 : Transfert de fichiers

Dans ce scénario, le but est de transférer des fichiers d'un dossier distant vers un dossier local. On dispose de 100 fichiers de 100ko chacun. Le serveur ne permet pas de télécharger plusieurs fichiers à la fois. Avec RMI, on doit donc récupérer les fichiers un par un. Avec l'agent mobile, on envoie un agent sur le serveur qui récupère les fichiers un par un, les compresse dans un zip, et revient avec en un seul aller-retour.

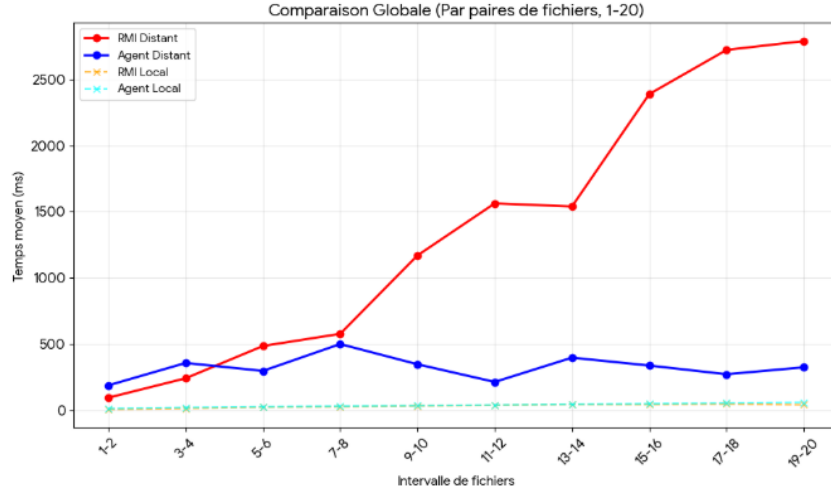


FIGURE 3 – RMI vs Agent appliqué à du transfert de fichiers

TABLE 2 – Scénario Fichiers (Distant)

N	RMI (ms)	Agent (ms)	Vainqueur	Facteur
1	~35	~200	RMI	$\times 5.7$
10	~1 380	~250	Agent	$\times 5.5$
50	~2 520	~700	Agent	$\times 3.6$
70	~3 200	~660	Agent	$\times 4.8$
100	~6 247	~509	Agent	$\times 12.2$

4 Analyse et Interprétation

- On observe bien un point de bascule très précoce, situé entre $N = 2$ et $N = 5$. En dessous de ce seuil, RMI est plus performant car le coût de migration de l'agent n'est pas amorti.
- Le temps d'exécution de l'agent mobile reste quasi constant, indépendamment du volume de données traité. Une fois sur place, l'agent effectue uniquement des accès mémoire locaux, dont le coût est négligeable face à la latence réseau. À l'inverse, le temps d'exécution du RMI croît linéairement avec N , chaque appel distant impliquant un aller-retour réseau.
- Il faut noter une variabilité liée à l'état du réseau. Nous avons essayé de pallier à ce problème en faisant une moyenne de 6 tentatives par N (nombre de fichiers ou de lectures CSV). Nous avons également pris en compte l'effet de chauffe de la JVM, et choisi de commencer tous nos tests par 4 allers-retours non comptabilisés.
- Nous avons fait le choix de ne pas inclure les `lookup()` dans le temps d'un appel RMI, ce qui ne change pas l'allure de la courbe ni le moment où les courbes se croisent.
- Même en testant sur un même ordinateur, l'utilisation de RMI fait appel à la pile TCP/IP, ce qui explique la croissance linéaire du temps que prend RMI à effectuer un scénario (courbe orange sur la figure 1).

5 Conclusion

Les résultats obtenus confirment ceux de l'article OOPSLA (1999), tout en montrant un point de bascule plus précoce. En conclusion, le RMI est adapté aux interactions ponctuelles et légères, tandis que les agents mobiles sont particulièrement efficaces pour les traitements distribués intensifs en données, en réduisant drastiquement le nombre de communications réseau.