

Étude comparative des clients HTTP synchrones (RestTemplate, Feign, WebClient) avec Eureka et Consul

Étudiant Ingénieur – Génie Informatique

1 Introduction

Dans une architecture microservices, la communication synchrone entre services est un élément clé de performance et de maintenabilité. Ce travail vise à comparer trois clients HTTP de Spring Boot : **RestTemplate**, **Feign** et **WebClient**, en utilisant deux mécanismes de découverte de services : **Eureka** et **Consul**.

2 Environnement de test

Deux microservices ont été implémentés :

- **Service Voiture** : API REST simulée en mémoire
- **Service Client** : consommation de l'API via les trois clients HTTP

Les tests de charge ont été réalisés avec JMeter pour 10, 50, 100, 200 et 500 utilisateurs simultanés. Les métriques mesurées sont : temps moyen, P95, débit, CPU et mémoire.

3 Résultats de performance

3.1 Performances — Eureka et Consul

Users	Discovery	Métrique	Rest	Feign	WebClient
10	Eureka	Temps moyen (ms)	17	18	13
		P95 (ms)	28	59	11
		Débit (req/s)	0.05	0.19	1.1
	Consul	Temps moyen (ms)	13	19	12
		P95 (ms)	16	59	12
		Débit (req/s)	1.1	0.02	1.1
50	Eureka	Temps moyen (ms)	12	26	9
		P95 (ms)	25	50	17
		Débit (req/s)	0.29	0.4	5.1
	Consul	Temps moyen (ms)	12	19	10
		P95 (ms)	18	45	18
		Débit (req/s)	5.1	0.07	5.1
100	Eureka	Temps moyen (ms)	11	9	9
		P95 (ms)	22	13	21
		Débit (req/s)	0.5	1.8	10.1
	Consul	Temps moyen (ms)	12	9	8
		P95 (ms)	17	16	21
		Débit (req/s)	10.1	0.18	10.1
200	Eureka	Temps moyen (ms)	10	7	8
		P95 (ms)	21	10	14
		Débit (req/s)	1.2	20.1	20
	Consul	Temps moyen (ms)	10	8	8
		P95 (ms)	14	19	11
		Débit (req/s)	20	1.16	20.1

500	Eureka	Temps moyen (ms)	11	7	8
		P95 (ms)	21	9	15
		Débit (req/s)	1.2	20.1	20.1
Consul	Temps moyen (ms)	11	7	8	
	P95 (ms)	14	10	17	
	Débit (req/s)	20	1.16	20.1	

4 Consommation CPU et Mémoire

Discovery	Client	CPU	Mémoire (MB)
Eureka	RestTemplate	~0%	220
	Feign	~0%	198.5
	WebClient	~0%	180
Consul	RestTemplate	~0%	253
	Feign	~0%	240
	WebClient	~0%	228

5 Simplicité et maintenabilité

Critère	RestTemplate	Feign	WebClient
Configuration initiale	Simple	Moyenne	Moyenne à élevée
Lignes de code	Élevé	Faible	Moyen
Complexité globale	Faible	Faible à moyenne	Élevée

6 Conclusion

Les résultats montrent que **WebClient** offre les meilleures performances globales en termes de latence et de débit, avec une consommation mémoire réduite. **Feign** se distingue par sa simplicité et sa maintenabilité, ce qui en fait un excellent choix pour les appels synchrones en microservices Spring. Enfin, l'impact du service discovery reste limité sur la latence, mais influence la stabilité du débit.