
Брокер кластерного між серверного балансування мережевого трафіку та навантаження

Підготував: Войлов Богдан Станіславович, ІПЗ-4

Науковий керівник: Бабич Т.А.

Мета роботи та завдання

Метою роботи є проектування та реалізація брокера кластерного міжсерверного балансування, який дозволить ефективно розподіляти навантаження між серверами.

Балансування з round-robin

Ідентифікатор Запиту	Навантаження на сервер
1	80%
2	30%
3	60%

Ідентифікатор Запиту	Ідентифікатор Сервера	Навантаження на сервер
1	A	80%
2	B	30%
3	A	80%+60%=120%

Метод найменшої кількості з'єднань

Ідентифікатор з'єднання	Тривалість з'єднання	Час встановлення з'єднання	Навантаження на сервер
1	60 секунд	-	10%
2	60 секунд	через 1 секунду після з'єднання 1	10%
3	60 секунд	через 1 секунду після з'єднання 2	80%
4	60 секунд	через 1 секунду після з'єднання 3	40%

Метод найменшої кількості з'єднань

Ідентифікатор з'єднання	Ідентифікатор сервера	Кількість з'єднань у момент розподілу	Навантаження на сервер після розподілу
1	A	A=0, B=0	10%
2	B	A=1, B=0	80%
3	A	A=1, B=1	10%+10%=20%
4	B	A=2, B=1	80%+40%=120%

Ресурсний метод

Ідентифікатор сервера	Навантаження на CPU	Навантаження на оперативну пам'ять	Навантаження на диск
A	30%	30%	90%
B	50%	70%	10%

Ідентифікатор запиту	Навантаження на CPU	Навантаження на оперативну пам'ять	Навантаження на диск
1	40%	40%	0%
2	40%	10%	30%

Варіанти розподілу

Ідентифікатор запиту	Ідентифікатор сервера	Навантаження на CPU	Навантаження на оперативну пам'ять	Навантаження на диск
1	A	$30\%+40\%=70\%$	$30\%+40\%=70\%$	$90\%+0\%=90\%$
2	B	$50\%+40\%=90\%$	$70\%+10\%=80\%$	$30\%+10\%=40\%$

Ідентифікатор запиту	Ідентифікатор сервера	Навантаження на CPU	Навантаження на оперативну пам'ять	Навантаження на диск
1	B	$30\%+40\%=70\%$	$10\%+30\%=40\%$	$90\%+30\%=120\%$
2	A	$50\%+40\%=90\%$	$70\%+40\%=110\%$	$0\%+10\%=10\%$

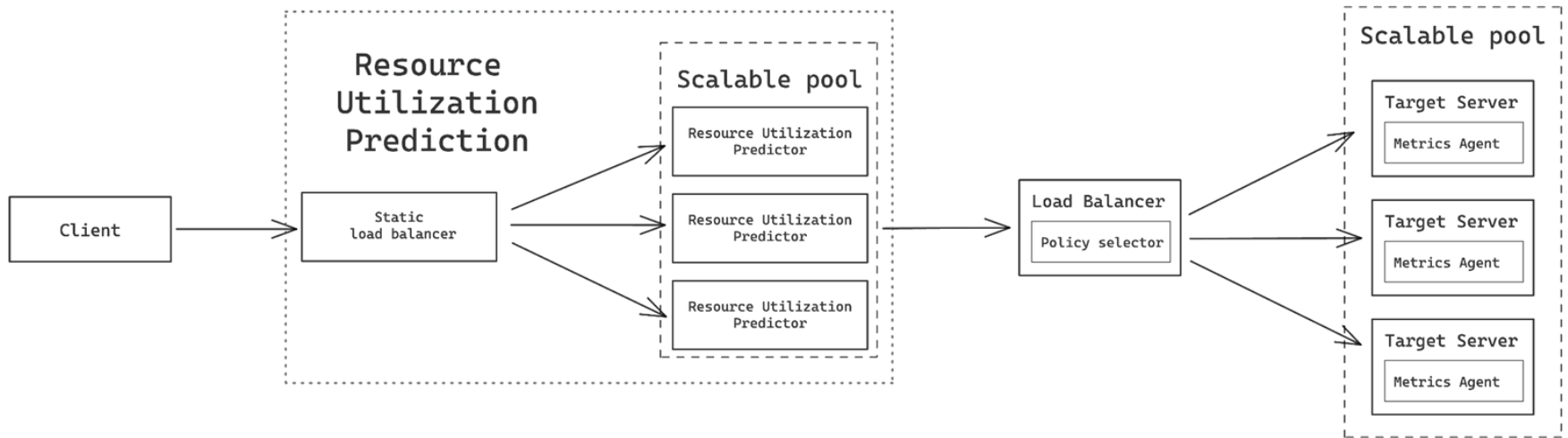
Вимоги до балансувальника

1. Висока доступність
2. Висока ефективність
3. Масштабованість
4. Надійність
5. Гнучкість і розширюваність
6. Підтримка різних протоколів: брокер підтримує різні протоколи для взаємодії з серверами та клієнтами.
7. Шифрування
8. Безпека
9. Підтримка різних середовищ

Використані технології

- Go - це мова програмування з відкритим кодом, яка дозволяє легко створювати просте, надійне та ефективне програмне забезпечення.
 - fasthttp - бібліотека для обробки http протоколу, розроблена для деяких високопродуктивних крайових випадків
 - Docker - це платформа, розроблена, щоб допомогти розробникам створювати, поширювати та запускати програми.
 - stress - інструмент для навантаження на системи стрес-тестування в linux
 - btest - утиліта тестування рівня блоку для Linux для створення синтетичних випадкових/послідовних шаблонів вводу-виводу
-

Архітектура



Алгоритм балансування

1. Обрати цільові сервери що задовольняють мінімальним пороговим значенням
2. На основі даних про стан сервера з агентів та прогнозу навантаження обчислити загальну оцінку
3. Перенаправити запит до сервера що має найменшу оцінку

Висновки