

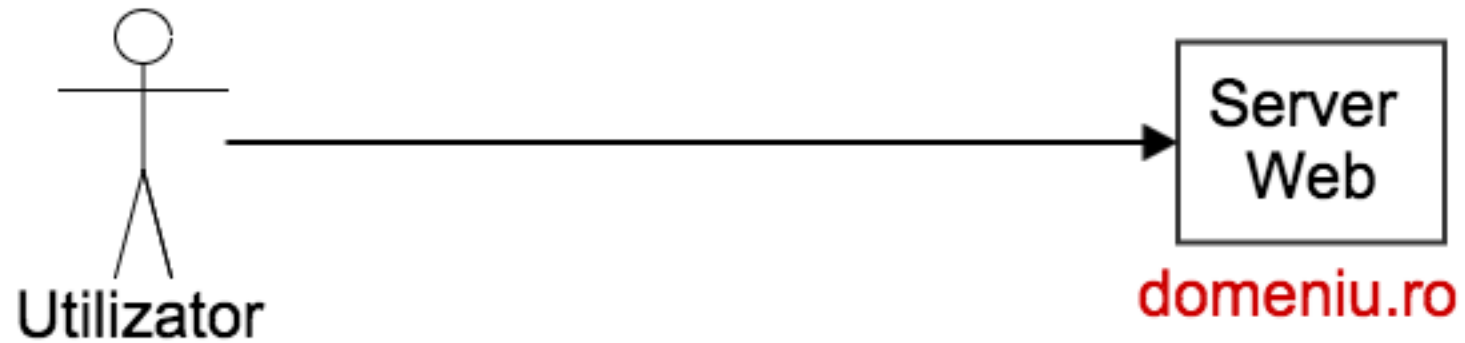
APLICATII DISTRIBUITE IN CLOUD

Load balancers. S3.



Load balancer

Scenariu



Serverul web poate servi X utilizatori simultan. Cum putem servi $X+1$ utilizatori?

Scenariu

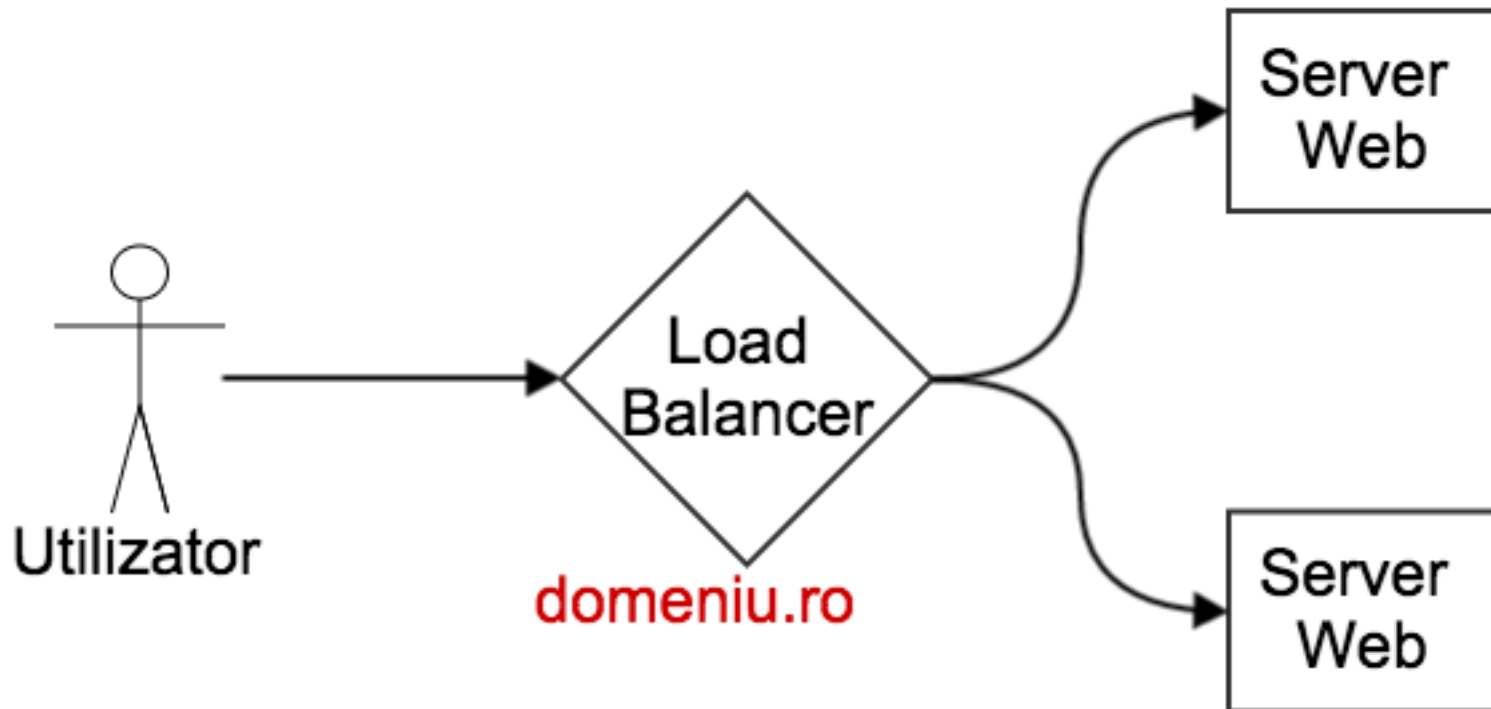
Adaugam inca un server



Al doilea server este functional dar nu poate folosi acelasi domain name.

Scenariu

Solutie: Alta masina care distribuie cererile



Numele de domeniu este folosit de LoadBalancer . Ambele servere sunt accesibile prin intermediul lui.

Load Balancer

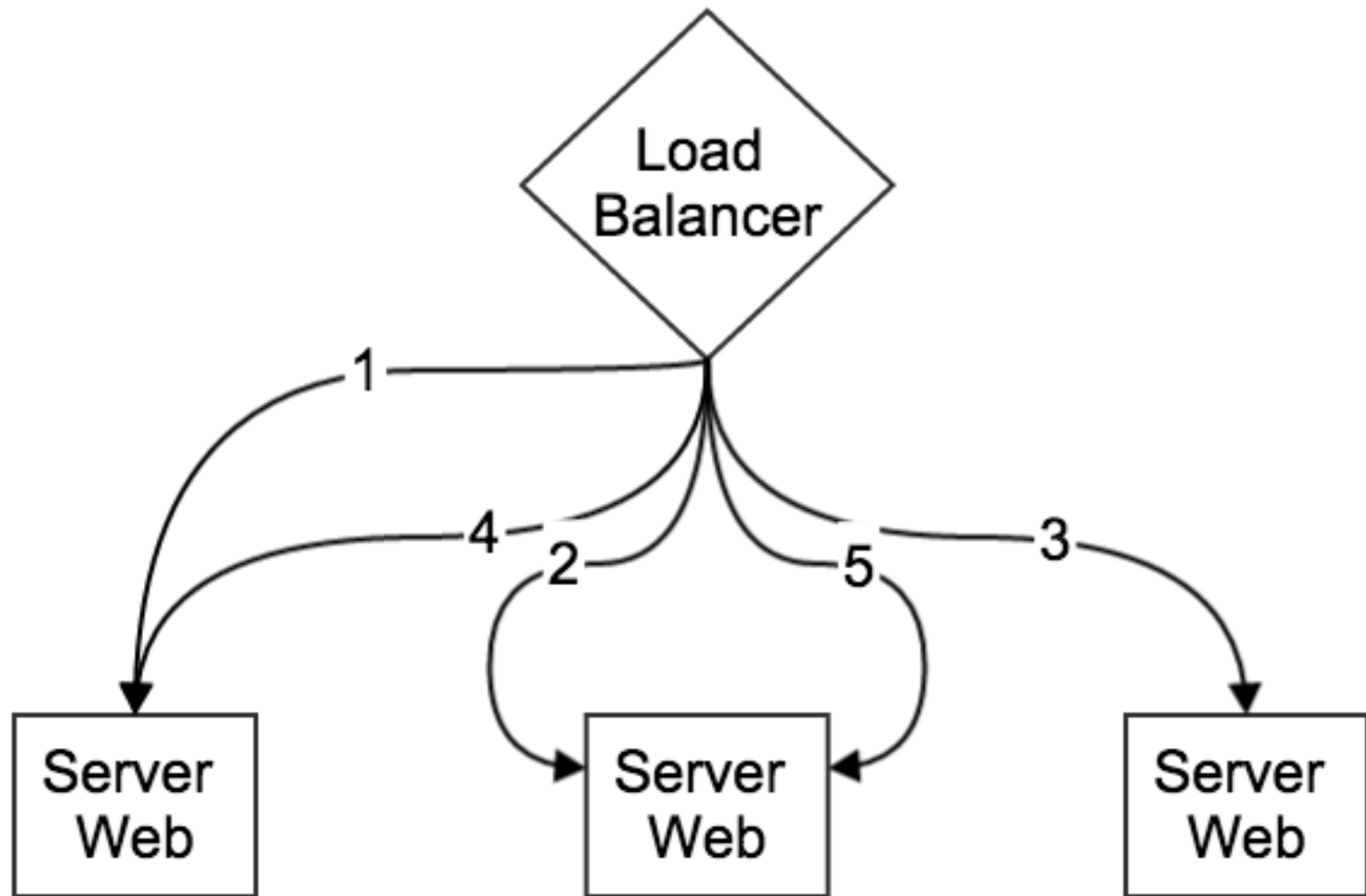
- Masina pe care ruleaza load balancer-ul are in general mai multe interfete de retea (astfel poate avea mai multe conexiuni concurente).
- Din cauza ca nu el face procesarea, masina pe care ruleaza poate avea resurse limitate.
- Daca serverele web sunt in internetul public, load balancer-ul poate sa redirecteze cererea direct intr-un server fara sa tina el conexiunea deschisa.

Load Balancer: Algoritmi

1. Round robin

- Cererile sunt distribuite uniform dupa formula:
 $\text{cerere}_x \rightarrow \text{masina}_x \bmod n$, unde 'x' este numarul de ordine al cererii si 'n' este numarul de servere.
- Presupune ca serverele nu mentin date despre cererile anterioare. Orice server poate procesa orice cerere in orice moment.

Load Balancer: Round Robin: Exemplu

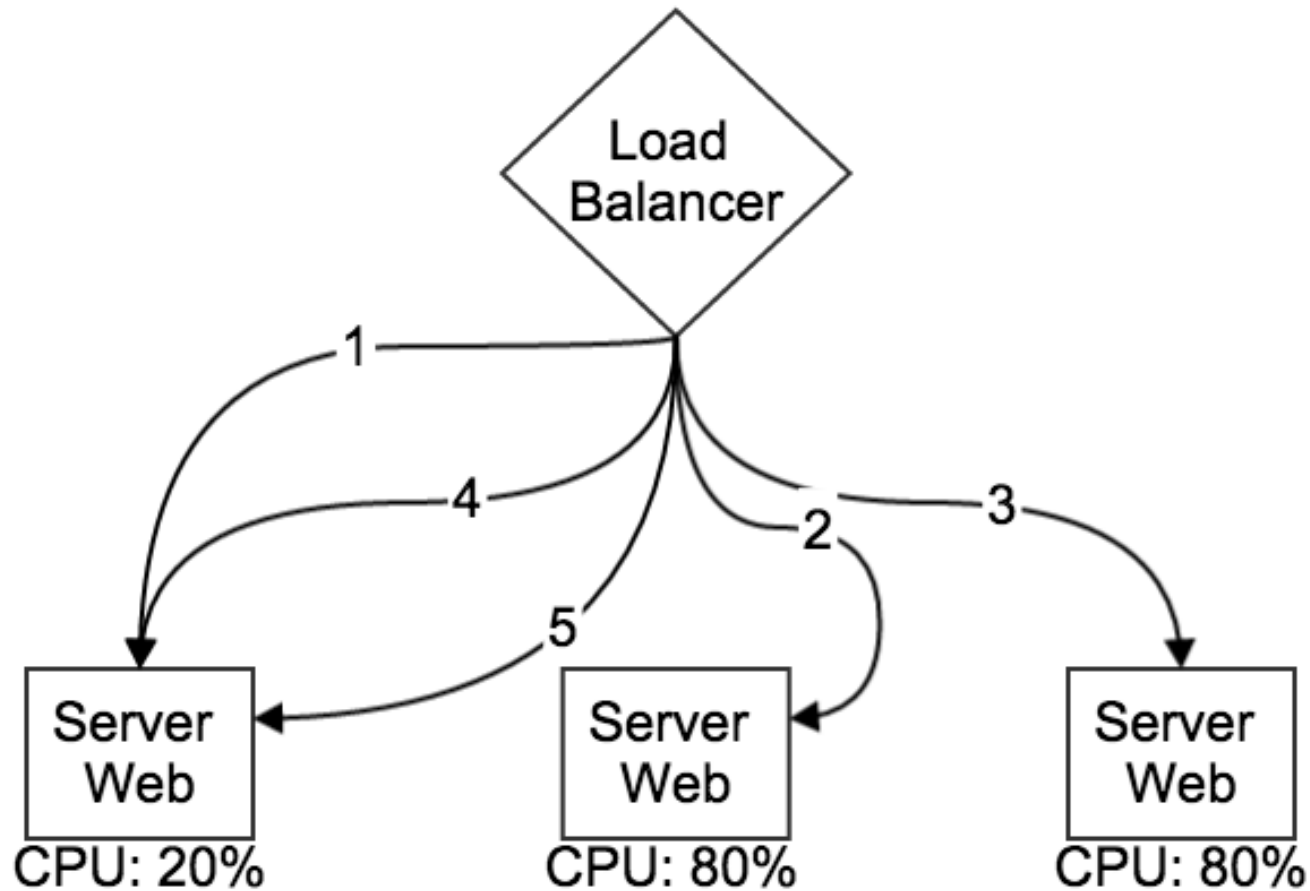


Load Balancer: Algoritmi

2. In functie de utilizarea resurselor

- Cererea este procesata de serverul care are factorul de incarcare cel mai mic (cpu, io, ram etc).
- Se foloseste cand timpul/resursele necesare procesarilor variaza mult in functie de tipul de cerere.
- Presupune ca serverele nu mentin date despre cererile anterioare. Orice server poate procesa orice cerere in orice moment.

Load Balancer: Resurse: Exemplu



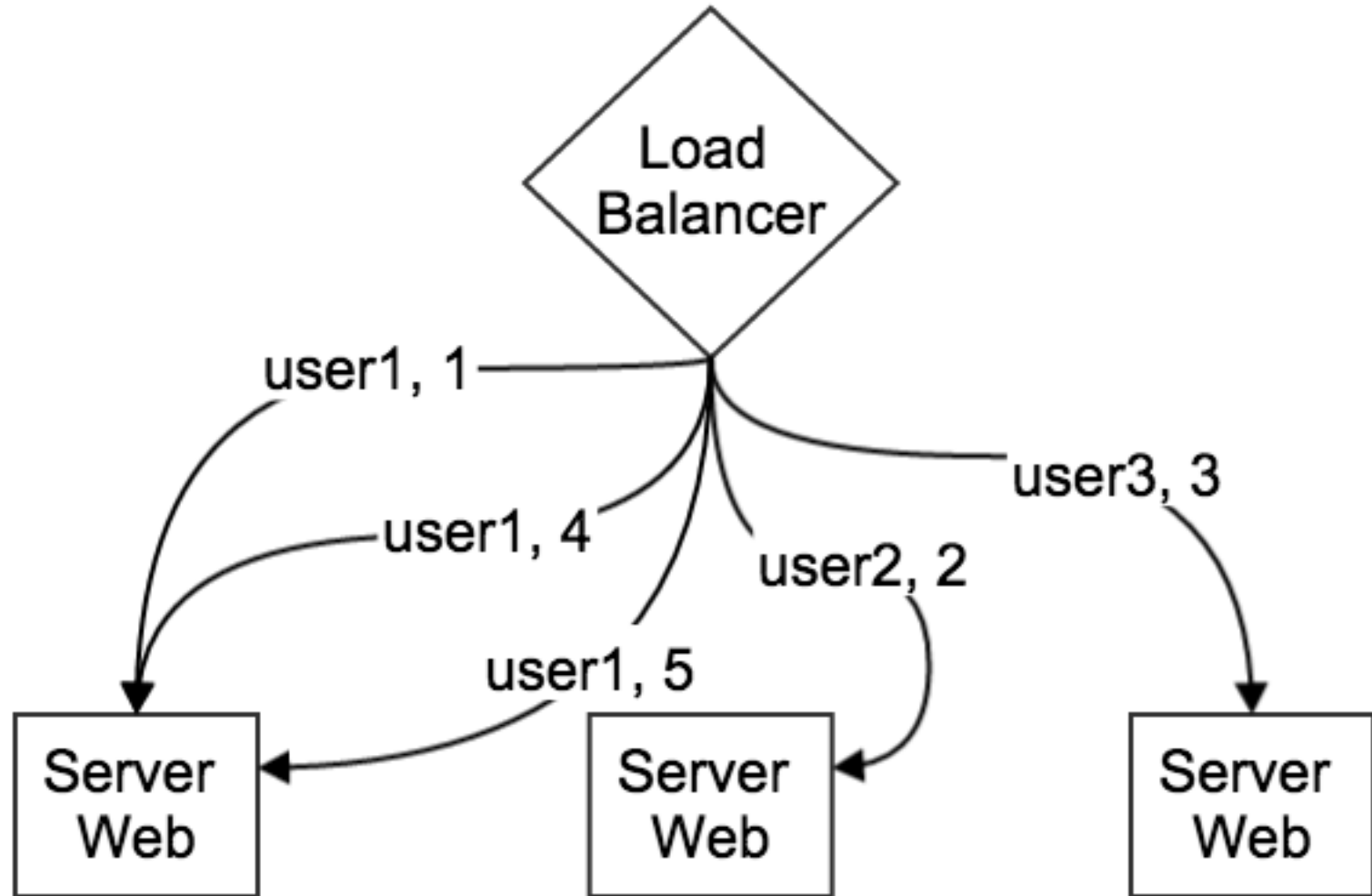
Cererile 2 si 3 au nevoie de multa procesare, iar 1, 4 si 5 de putina.

Load Balancer: Algoritmi

Sticky sessions

- Daca o cerere are asociata un id de sesiune, cererea merge la serverul web care a creat sesiunea respectiva. Daca nu are asociata un id de sesiune se foloseste alt algoritm.
- Cererile nu sunt distribuite uniform.
- De exemplu un magazin virtual trebuie sa foloseasca sticky sessions, altfel cosul de cumparaturi nu va functiona.

Load Balancer: Sticky Session: Exemplu



Load Balancer: Algoritmi

Eliminarea nevoii de sticky sessions

Sesiunile pot fi stocate intr-o baza de date (nu pe serverul web) iar serverul web extrage informatiile de acolo cand cererea are asociata un id de sesiune. Astfel nu este nevoie de sticky sessions dar trebuie luat in considerare ca pentru fiecare cerere venita de la utilizator, serverul web interogheaza baza de date.

Load Balancer: Consistent

- Load balancer-ul verifica permanent starea serverelor web (healthcheck).
- Daca un server web se defecteaza, load balancer-ul il scoate din lista lui de clienti.

Load Balancer in AWS

ELB: **E**lastic **L**oad **B**alancer

- Gestionat de AWS
- Scaleaza automat in functie de trafic
- Suport pentru IPv6
- Suport pentru Sticky Session



S3

S3: Simple Storage Service

- ❑ Sistem de stocare performant si scalabil expus ca serviciu web.
- ❑ Stocare nelimitata.
- ❑ Viteza de transfer foarte mare intre el si masinile EC2.
- ❑ Intr-un cont se pot crea mai multe bucket-uri pentru a separa datele.

S3: Simple Storage Service

- ❑ Stocare <Cheie, Valoare>. Valoare poate fi un fisier de 5TB
- ❑ Disponibilitate foarte mare: 99,9999999999%
- ❑ Versionare automata pentru fisiere.

S3

Operatii cu S3:

- Adaugarea unei valori: `adauga(<"cheie1", "1 23">)`
- Stergerea unei valori: `sterge("prefix/cheie2")`
- Citirea unei valori: `val = citeste("prefix/cheie3")`.
- Listarea cheilor cu prefix comun: `listeaza("prefix/")`
 - ▣ `prefix/cheie2`
 - ▣ `prefix/cheie3`

Obs: Folosind listarea cheilor cu prefix comun putem simula lucrul cu directoare.

Intrebari

