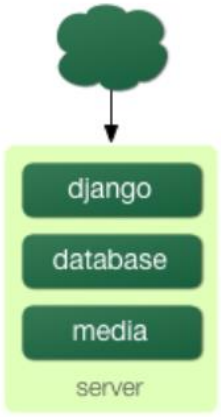
ISA/MRS Tim11 PoC

Pokretanje na jednom serveru

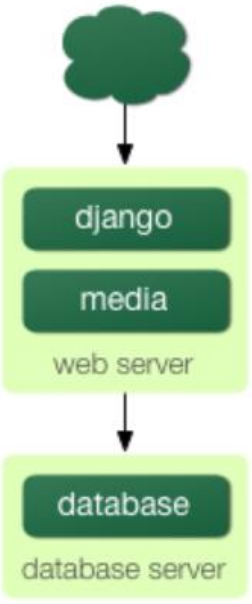
Većina sajtova počinje na jednom serveru, sa arhitekturom kao na slici 1. Međutm, kako saobraćaj raste, doći će do “takmičenja za resurse” (*resource contention*) između različitih delova softvera.



Slika1

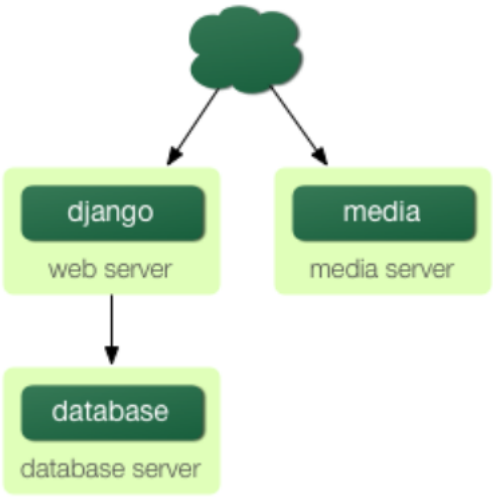
Odvojen server baze podataka

Proces odvanja servera baze podataka u Djangu je veoma lak. Potrebno je promeniti DATABASE_HOST u podešavanjima na IP adresu ili DNS ime servera baze podataka.



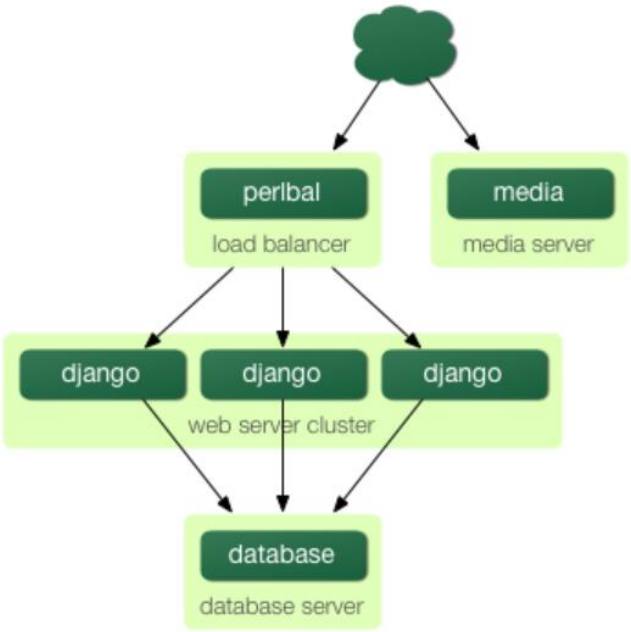
Odvojen *media* server

Sledeći korak je da se odvoji *media* (sve ono što nije generisano od strane Django view-a) na poseban server



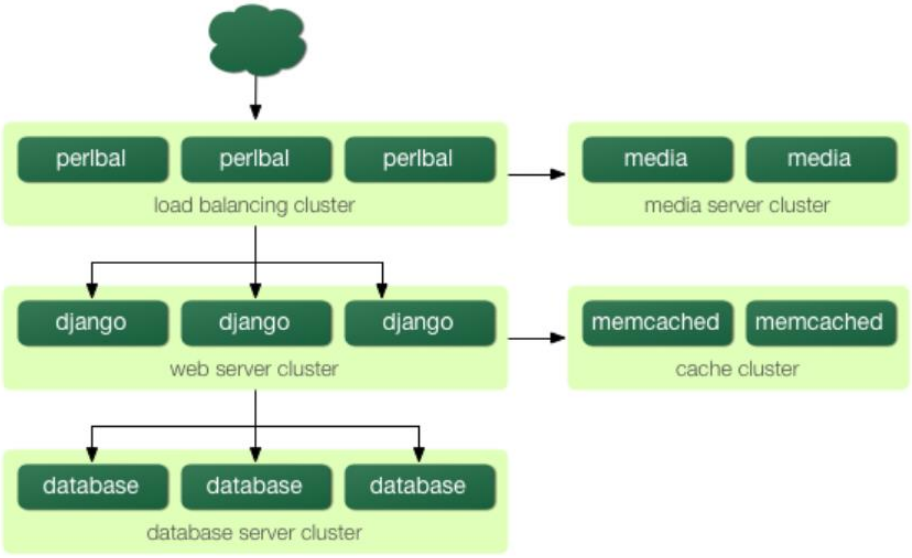
Impementacija Load Balacing-a i Redudantnosti

Ukoliko dodamo redudantne servere, povećaćemo kapacitet I pouzdanost. Tako da ako otkaže jedan server ostali će raditi. Međutim potreban nam je softver koji će raspodeliti saobraćaj na više servera – *load balancer*.



Sledeći koraci su uglavnom izvodi prethodnih:

- Ako su nam potrebne veće performanse baze podataka dodaćemo još servera. MySQL ima ugrađenu replikaciju.
- Ako nam jedan *load balancer* nije dovoljan, dodajemo više mašina i povezujemo da rade po *Round Robin* sistemu.
- Isto važi i za *media* servere
- Dodajemo odvojene *cache* servere



Isključiti Keep-Alive

Keep-Alive je svojstvo HTTP-a koji koje omogućava da više HTTP zahteva budu usluženi preko jedne TCP konekcije. Ovo je dobro na prvi pogled, ali može smanjiti perfomanse. Uglavnom se zahtevi jednog korisnika ne šalju toliko često(na svakih 10 sekundi npr). Ovo ostavlja neaktivan HTTP server da čeka na sledeći keep-alive zahtev, i troši RAM, koji bi aktivan mogao da koristi.

Koristiti *Memcached*

Memcached je distribuiran sitem opšte namene za keširanje memorije. Smanjuje broj čitanja sa eksternih izvora.