

Hopota Raul-Sorin

An IV, AC, IS

Tema 2 DATC

Microsoft Azure SQL Elastic Pools

Developerii ce construiesc aplicații tip SaaS(Software as a Service si se refera practic la orice aplicație conectata la Cloud) se folosesc de cantități uriașe de date cuprinse in mai multe baze de date. In general la aceste tipuri de aplicații se pot folosi in principal doua modele din punct de vedere a repartizării bazelor de date: o baza de date pentru fiecare client sau o baza de date comuna pentru toți clienții unei aplicații. Bineînțeles ca exista avantaje si dezavantaje la folosirea unuia dintre modele in favoarea celuiilalt.

Unul din dezavantajele folosirii unei baze de date pentru toți clienții ar fi problema izolării folosirii de resurse a unui client de ceilalți clienți. Rezulta faptul ca pot apărea situații in care un singur client consuma toate resursele din baza de date deci si resursele celorlalți ușeri ai aplicației respective. Alt dezavantaj ar fi in momentul in care este nevoie de un back-up sau un restore la unul dintre userii aplicației moment in care ar trebui oprit serviciul pentru toți userii din acea baza de date pentru a putea efectua acest back-up. Daca am avea cate o baza de date pentru fiecare client atunci am opri serviciul doar pentru acea baza de date si doar clientul pentru care are nevoie de back-up ar fi afectat.

Cel mai important dezavantaj in folosirea unei baze de date pentru cate un client este faptul ca daca dorești sa construiești o aplicație astfel in cat performanta sa fie mare atunci trebuie sa prevezi un anumit volum de resurse care sa fie de ajuns pentru momentele in care userul are nevoie de un anumit număr maxim de resurse la un moment dat in timp. Deci trebuie sa alocam mereu pentru numărul maxim de resurse de care are nevoie un user si asta se numește supra aprovizionare cu resurse si nu este deloc eficienta din punct de vedere al costului pentru ca ar trebui sa plătim pentru un număr maxim de resurse pe care user-ul nu le folosește decât la un anumit moment in timp.

Aici intervine noțiunea de Elastic pools care ne lasă sa beneficiem de toate calitățile aduse de „izolare” si in același timp sa împărțim numărul total de resurse al tuturor bazelor de date intre acestea. Elastic pool aduce laolaltă mai multe baze de date intr-un „pool” in care toate resursele sunt împărțite si deci volumul de resurse va fi mult redus, de asemenea si costul.

O noțiune importanta ce apare in acest context este aceea de DTU. Un DTU(Data Troughput Unit) este o unitate de măsura pentru performanta unui nivel de servicii si reprezintă o combinație a mai multor caracteristici al bazelor de date. Fiecare nivel de serviciu are un număr de DTU-uri atribuit acestuia ca o modalitate ușoară de a compara performantele unui nivel de serviciu fata de altul. DTU-urile oferă o modalitate de a descrie capacitatea relativa a unui nivel de performanta al bazelor de date Basic, Standard si Premium. DTU se

bazează pe o amestecătură de măsurări a procesorului, memoriei , a citirii si scrierii de date(I/O). Pe măsură ce DTU-urile cresc puterea oferita de nivelul de performanta creste. De exemplu, un nivel de performanta cu 5 DTU-uri are o putere de cinci ori mai mare decât un nivel de performanta cu 1 DTU.

Un dezvoltator poate configura numărul de resurse pentru un „pool” folosindu-se de modelul de achiziție bazat pe DTU. Numărul total de resurse este controlat in totalitate de bugetul dezvoltatorului. Cu aceste sistem dezvoltatorul doar adaugă bazele de date in „pool” si setează numărul minim si numărul maxim de resurse alocate tuturor bazelor de date(numărul minim si maxim de DTU-uri) si apoi setează resursele acestui „pool”, totul in limita bugetului pe care îl are.

In cadrul unui „pool”, bazelor de date individuale li se da flexibilitatea de a scala automat intre valorile parametrilor stabiliți. In cazul unei sarcini grele o baza de date poate consuma mai multe resurse pentru a satisface cererea. Pe de alta parte bazele de date cu sarcini ușoare consuma mai puțin si bazele de date fără sarcina nu consuma resurse. Alocând resurse pentru întregul „pool”, mai degrabă decât pentru bazele de date unice, poate simplifica sarcinile de gestionare.

Un plus îl reprezintă faptul ca se va știi valoare unui buget previzibil pentru „pool”. De asemenea se pot adăuga resurse suplimentare pentru un „pool” existent fără a întrerupe funcționarea unei baze de date, cu excepția faptului ca este posibil ca bazele de date sa trebuiască sa fie mutate pentru a furniza resurse suplimentare de calcul pentru noul total de DTU. Similar daca resursele suplimentare adăugate nu mai sunt necesare, ele pot fi eliminate dintr-un pool in orice moment. Se pot de asemenea adăuga sau elimina/muta baze de date la orice moment, de exemplu, daca o baza de date este puțin probabil sa utilizeze resurse aceasta poate fi eliminata daca nu mai este de folos sau mutata altfel.

Aceste „pools” sunt recomandate a fi folosite in cazul unui număr mare de baze de date cu pattern-uri specifice de utilizare. Pentru o baza de date data acest pattern este caracterizat de o medie scăzută a folosirii. Cu cat mai multe baze de date se pot adauga unui pool cu atât mai mari vor fi economiile făcute.

In ceea ce privește prețul, acesta variaza in funcție de numărul de DTU-uri al pool-ului. In timp ce prețul unității DTU pentru un pool este de 1.5 ori mai mare decât prețul unității DTU pentru o singura baza de date, DTU-urile pot fi împărțite de către mai multe baze de date si sunt necesare mai puține DTU-uri per total. Aceste distincții in ceea ce privește stabilirea preturilor si partajarea DTU-urilor reprezintă baza potențialului de reducere a preturilor pe care îl poate oferi un pool.

Cea mai buna dimensiune pentru un pool este determinata de ansamblul resurselor necesare pentru toate bazele de date din acel pool. Aceasta dimensiune este determinata de numărul maxim de resurse utilizat de toate bazele de date din pool si de numărul maxim de octeți de stocare utilizat de toate bazele de date din pool.

Deci, aceasta soluție oferita de Microsoft pentru gestiunea si folosirea intr-un mod cat de optim posibil a bazelor de date este una foarte ingenioasa si foarte folositoare in același timp si din punctul meu de vedere ar trebui sa fie folosita pentru aplicații ce au un număr foarte mare de utilizatori sau folosesc un număr mare de resurse.