

Platforme za izvršavanje u oblaku pružaju različite servise (eng. services) korisnicima koristeći razne modele isporuke usluga. Savremena istraživanja ovih modela dovela su do široke primene bezserverskog izvršavanja (eng. serverless), paradigme u kojoj se softver sastoji od funkcija — brzih, ponovno iskoristivih jedinica koda koje se izvršavaju unutar izolovanih, virtualizovanih okruženja. Vodeće platforme za izvršavanje u oblaku, uključujući Amazon Web Services (AWS), Microsoft Azure i Google Cloud, izveštavaju da značajan deo njihovih korisnika koristi bezserverska rešenja.

Većina pružalaca servisa izvršavanja u oblaku primenjuje model naplate po principu „plati koliko koristiš”. Neefikasno korišćenje računarskih resursa, naročito procesorskih jezgara i radne memorije, koji predstavljaju dva najskuplja resursa, dovodi do značajnog povećanja troškova. Dodatno, potreba za virtualizacijom negativno utiče na vreme inicijalizacije i povećava potrošnju procesorskog vremena i radne memorije. Izolovana bezserverska okruženja se tipično implementiraju nad okruženjima koja zahtevaju velike količine radne memorije, kao na primer virtualne mašine za jezike Java, JavaScript ili Python i odgovarajući prateći radni okviri.

Savremene arhitekture za izvršavanje u oblaku koriste tehnike snimanja stanja inicijalizovanih okruženja u nastavljivom obliku (eng. Checkpoint/Restore). Ove tehnike omogućavaju optimizaciju iskorišćenja resursa, kao i deljenje koda i podataka između više okruženja za izvršavanje. Međutim, postojeća rešenja funkcionišu u fazi izgradnje aplikacija i, kao takva, ne omogućavaju ranu inicijalizaciju i deljenje podataka koji postaju dostupni u toku izvršavanja aplikacija. Takvi podaci se iznova obrađuju i dupliraju u svakom pojedinačnom izolovanom okruženju za izvršavanje.

Ova disertacija predstavlja Doss, sistem za snimanje i deljenje podataka koji omogućava snimanje i deljenje podataka tokom izvršavanja aplikacije. Doss direktno snima objekte koji sačinjuju podatke, bez njihove transformacije, u obliku ponovno iskoristivih i deljivih snimaka (eng. snapshot). Takav pristup omogućava Doss-u da postigne konstantno vreme deserijalizacije podataka, čime se značajno unapređuje vreme inicijalizacije aplikacija i smanjuje potrošnja računarskih resursa. Arhitektura sistema Doss omogućava deljenje snimaka između više instanci aplikacije, čime se eliminiše memorijsko zauzeće povezano sa ponovnom obradom i dupliranjem podataka i poboljšava vreme inicijalizacije aplikacije.

Implementacija sistema Doss, pod nazivom GraalDoss, realizovana je u programskom jeziku Java i integrisana u ekosistem GraalVM . GraalDoss je evaluiran pomoću 106 testova korektnosti i robustnosti, kao i korišćenjem novog skupa referentnih programa namenjenih testiranju aplikacija u oblaku. Rezultati evaluacije pokazuju konzistentno vreme deserijalizacije, uz brzinu serijalizacije uporedivu sa savremenim Java bibliotekama za JSON i binarnu serijalizaciju. U mikroservisnim veb aplikacijama, GraalDoss eliminiše memorijsko zauzeće privremenih skladišta podataka deljenjem snimaka popunjenih skladišta između instanci mikroservisa, čime se postiže povećanje odnosa brzine i memorijskog zauzeća sistema od 41% za osam instanci mikroservisa, uz smanjenje vremena odziva aplikacije od 34%. U aplikacijama koje koriste obradu prirodnih jezika, GraalDoss unapređuje vreme izvršavanja za šest redova veličine snimanjem modela za obradu teksta i rezultata njegove primene na ulazni tekst.