

Sistemi Distribuiti e Cloud Computing - A.A. 2020/21

Progetto A1: Un sistema di storage distribuito di tipo chiave-valore per l'edge computing

Docente: Valeria Cardellini

Dipartimento di Ingegneria Civile e Ingegneria Informatica

Università degli Studi di Roma "Tor Vergata"

cardellini@ing.uniroma2.it

Requisiti del progetto

Lo scopo del progetto è realizzare, nel linguaggio di programmazione Go, un **sistema di storage distribuito per edge computing**.

Il sistema proposto ha l'obiettivo di fornire un servizio di storage replicato di tipo chiave-valore in uno scenario di edge computing. L'edge computing è un paradigma di computazione [1] che estende il Cloud tradizionale con le funzionalità di computazione e storage offerte da nodi (dispositivi IoT, smartphone, micro data center) localizzati ai bordi della rete (ad es. in prossimità degli access point) e quindi vicino agli utenti. Le funzionalità di tali nodi edge, tipicamente caratterizzati da una limitata capacità computazionale e di storage, sono integrate dai più tradizionali servizi Cloud, forniti da data center localizzati nel core di Internet e pertanto lontani dagli utenti.

Tramite il sistema da realizzare, un'applicazione sviluppata per un dispositivo o per un sensore può memorizzare tramite chiamate RPC le operazioni di Put, Get, Del e Append sui valori associati alle chiavi. L'operazione Put(key, value) consente di memorizzare il valore key a cui è associata una chiave value, eventualmente sovrascrivendo il valore già presente. L'operazione Get(key) consente di leggere un valore a cui è associata la chiave key specificata. L'operazione Del(key) consente di rimuovere un valore a cui è associata la chiave key. L'operazione di Append(key, arg) consente di modificare un valore già memorizzato a cui è associata la chiave key, aggiungendo il valore specificato da arg.

Il sistema di storage distribuito deve soddisfare i requisiti funzionali e non funzionali elencati di seguito.

- Essere distribuito su molteplici nodi (eventualmente localizzati in un contesto urbano di Smart City), considerando che la latenza di rete tra i nodi usati dal sistema è non trascurabile. I nodi del sistema sono scoperti durante l'avvio del sistema di storage attraverso un servizio di registrazione (che può essere implementato anche in modo centralizzato) e possono essere soggetti a crash durante il funzionamento del sistema.
- Essere scalabile (rispetto al numero di utenti e di dati). Si richiede di adottare a tale scopo tecniche di replicazione dei dati su molteplici nodi del sistema. Si richiede inoltre che il modello di consistenza offerto dal sistema di storage sia almeno di tipo finale e la semantica degli errori di tipo at-least-once.

- Supportare l'integrazione con un servizio di storage Cloud allo scopo di incrementare la scalabilità rispetto ai dati, memorizzando ad esempio nel Cloud valori di grandi dimensioni e/o scarsamente acceduti.
- *Opzionale per gruppi composti da 2 studenti, obbligatorio per gruppi composti da 3 studenti* Essere tollerante a crash dei nodi server che compongono il sistema.
- *Opzionale per gruppi composti da 2 studenti, obbligatorio per gruppi composti da 3 studenti* Supportare la migrazione dei dati tra i nodi del sistema allo scopo di ridurre la latenza per i sensori o dispositivi che accedono ai dati.

Non avendo a disposizione una piattaforma di edge computing, i nodi edge possono essere eseguiti sul proprio laptop ad es. all'interno di container oppure su istanze EC2 di una data regione; in quest'ultimo caso, si consideri una diversa regione per il servizio Cloud utilizzato dall'applicazione, in modo da simulare una latenza di rete tra nodi edge e Cloud.

Si realizzi un testing completo delle funzionalità del sistema realizzato e si effettui un testing delle prestazioni misurando all'aumentare del numero di richieste il tempo di risposta del sistema nei seguenti casi:

1. carico di lavoro composto prettamente (85%) da operazioni di Get, per il restante 15% da operazioni di Put;
2. carico di lavoro composto da operazioni di Put (40%), operazioni di Append (20%), operazioni di Get (40%).

È possibile usare librerie e tool di supporto allo sviluppo del progetto; le librerie ed i tool usati devono essere esplicitamente indicati e brevemente descritti nella relazione e non devono essere sovrapposti con gli obiettivi del progetto.

Scelta e consegna del progetto

Il progetto è dimensionato per essere realizzato da un gruppo composto da **2 o 3** studenti. Essendo un progetto di tipo di A, il voto del progetto peserà il 50% della valutazione complessiva dell'esame.

Per poter sostenere l'esame nell'A.A. 2020/21, **entro il 24/8/2021** è necessario prenotarsi per il progetto, comunicando alla docente in una email avente come oggetto **[SDCC scelta progetto]** le seguenti informazioni:

- nome e cognome, email e numero di matricola dei componenti del gruppo;
- progetto scelto.

Nel caso in cui il numero di prenotazioni per il progetto scelto abbia raggiunto la soglia massima prevista, sarà necessario effettuare una nuova scelta tra i progetti ancora disponibili.

Eventuali modifiche nella scelta del progetto devono essere tempestivamente comunicate alla docente e con lei concordate. Non è possibile cambiare la tipologia del progetto.

Per ogni comunicazione via email è necessario specificare **[SDCC]** nell'oggetto della email. Il progetto è valido **solo** per l'A.A. 2020/21 e deve essere consegnato **entro il 30/11/2021**. La prova d'esame scritta deve essere superata **entro la sessione autunnale 2020/21**.

La consegna del progetto (nel caso in cui non si intenda partecipare al challenge) deve avvenire almeno 5 giorni lavorativi prima della data (da concordare con la docente) in cui si intende sostenere la discussione del progetto. La consegna del progetto consiste nell'invio alla docente di una email avente come oggetto **[SDCC consegna progetto]** e contenente un link ad uno spazio di Cloud storage o repository (e.g., Dropbox, Google Drive, GitHub) contenente:

1. il codice sorgente (opportunamente commentato);
2. relazione (in formato pdf);
3. un breve howto per l'installazione, la configurazione e l'esecuzione dell'applicazione.

La relazione contiene:

- la descrizione dell'architettura del sistema e dei suoi componenti;
- la descrizione delle scelte progettuali effettuate, opportunamente motivate;
- la descrizione dei principali aspetti relativi all'implementazione;
- la descrizione delle eventuali limitazioni riscontrate;
- l'indicazione della piattaforma software usata per lo sviluppo ed il testing del sistema (incluse eventuali librerie);
- i risultati del testing di prestazione opportunamente commentati.

Si consiglia di redarre la relazione in forma di articolo scientifico di lunghezza massima pari a 8 pagine, usando il formato ACM proceedings (<https://www.acm.org/publications/proceedings-template>) oppure il formato IEEE proceedings (https://www.ieee.org/conferences_events/conferences/publishing/templates.html).

Valutazione del progetto

I principali criteri di valutazione del progetto saranno:

1. rispondenza ai requisiti;
2. originalità;
3. organizzazione del codice (leggibilità, modularità, ...);
4. deployment del sistema;
5. organizzazione, chiarezza e completezza della relazione.

Riferimenti bibliografici

[1] W. Shi and S. Dustdar. The promise of edge computing. *IEEE Computer*, 49(5):78–81, 2016.