

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CAMPUS DI CESENA
SCUOLA DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

ESLS: ESLS SMART LIGHT SYSTEM

Elaborata nei corsi di:
Ingegneria dei Sistemi Software Adattativi Complessi
Smart City

Progetto di:
LUDOVICO DE NITTIS
EUGENIO SEVERI

ANNO ACCADEMICO 2016–2017

PAROLE CHIAVE

Bitcoin

IoT

Raspberry Pi

Vending Machine

Distributore Intelligente

Indice

Introduzione	vii
1 Analisi dei requisiti e del problema	1
1.1 Requisiti	1
1.2 Analisi dei requisiti	2
1.2.1 Estendibilità	2
1.2.2 Dinamicità	2
1.2.3 Alte prestazioni	3
1.2.4 Reattività	3
1.2.5 Scalabilità	4
1.2.6 Robustezza	5
1.2.7 Qualità del servizio adeguata	5
1.2.8 Facilità di gestione	6
1.2.9 Sicurezza	6
1.2.10 Adattabilità	6
1.3 Analisi del problema	7
2 Progettazione	9
2.1 Progettazione generale del sistema	9
2.2 Architettura complessiva	9
2.3 Controllo presenza auto	10
2.3.1 Comportamento	10

Introduzione

Capitolo 1

Analisi dei requisiti e del problema

1.1 Requisiti

Il sistema precedentemente introdotto necessita dei seguenti requisiti:

- estendibilità, ovvero deve essere possibile modificare, aggiungere o rimuovere dispositivi al sistema;
- dinamicità, ovvero deve essere possibile aggiungere, rimuovere o ri-collocare i dispositivi Raspberry sul territorio, senza interrompere il servizio nel complesso;
- ad alte prestazioni, cioè in grado di processare le molte richieste contemporanee rapidamente;
- reattività, cioè in grado di gestire le richieste in maniera non bloccante, in modo da essere sempre in grado di accettarne di nuove;
- scalabilità, in modo da far fronte all'elevato numero di richieste, direttamente proporzionali al numero di Raspberry nel sistema;
- robustezza, ovvero tolleranza ai guasti: il malfunzionamento di un Raspberry non deve compromettere il resto del sistema, ma solo le risorse direttamente collegate ad esso;

- qualità del servizio adeguata al tipo di applicazione: si deve minimizzare l'impatto dei guasti sugli utenti finali;
- facilità di gestione da parte degli amministratori: tutti i dispositivi devono essere gestiti in maniera centralizzata, senza doversi recare fisicamente ad ognuno;
- sicurezza: data la natura del sistema, sarà opportuno progettare e implementarlo secondo opportune best practices di sicurezza;
- adattabilità al funzionamento su diverse tipologie di rete, con throughput e latenza variabili.

1.2 Analisi dei requisiti

Verranno ora analizzati i requisiti presentati.

1.2.1 Estendibilità

Il sistema deve essere realizzato in modo tale da poter essere modificato nel tempo aggiungendo, rimuovendo o modificando la configurazione dei dispositivi, senza costringere il committente a smantellare il sistema corrente e crearne uno nuovo. Trattandosi di un sistema distribuito, deve inoltre essere possibile aggiornare il software dei vari componenti da remoto, ad eccezione dei casi in cui sia richiesto un intervento hardware. Anche per questo motivo si è scelta come piattaforma Raspberry Pi, che esegue un sistema operativo Linux completo, facilmente riconfigurabile. Ogni Raspberry può gestire un numero variabile di lampioni, in base alla topologia del territorio e alle esigenze del committente.

1.2.2 Dinamicità

Il sistema deve consentire l'aggiunta, rimozione e ricollocamento dei dispositivi Raspberry sul territorio, oltre ai relativi lampioni, senza dover spegnere il sistema nel complesso, in modo da non causare interruzioni del servizio. Il sistema deve essere quindi adattabile ai cambiamenti delle configurazioni. Anche per questo motivo saranno definite interfacce REST.

1.2.3 Alte prestazioni

Il sistema dovrà essere performante, ovvero in grado di processare le molte richieste ad alta velocità. Questo è possibile grazie all'uso delle interfacce web e alla scalabilità dei servizi cloud. Considerando un numero di Raspberry pari a circa 2000, le richieste che verranno inviate saranno così composte:

- notifiche di accensione/spengimento (due al giorno);
- notifiche di attivazione e disattivazione modalità notturna (due al giorno);
- notifiche di rilevamento del passaggio di automobili (5 lampioni stimati per ogni Raspberry * 20 auto rilevate stimate * 2 transizioni di aumento e riduzione della luminosità).

Si stima quindi un numero di richieste giornaliere inviate nell'ordine delle 400000 unità $[2000 * (2 + 2 + 200)]$. Nei momenti di picco, è verosimile pensare a qualche centinaio di richieste contemporanee, che si verificano principalmente quando molti lampioni si accendono o spengono contemporaneamente, mentre nei momenti di traffico dati minore (durante il giorno) le richieste scenderebbero praticamente a zero. Se il numero di tali richieste aumentasse al punto da causare problemi al server nella gestione, sarebbe possibile, lato Raspberry, non inviare le richieste nell'esatto momento in cui si verificano, ma con un ritardo casuale compreso in un intervallo selezionabile, in modo da distribuire meglio le richieste nel tempo. In questo modo le informazioni sul server potrebbero non essere aggiornate all'istante, ma dopo qualche secondo o minuto. Questo non sarebbe un problema, poiché quelle informazioni hanno uno scopo esclusivamente statistico, quindi non è necessaria la loro immediata disponibilità.

1.2.4 Reattività

L'aspetto principale che influenza la reattività del sistema è la latenza di trasmissione ed elaborazione delle informazioni scambiate tra i dispositivi. Anche per questo motivo saranno definite interfacce REST per la quasi totalità delle comunicazioni. Poiché la maggior parte del traffico dati è relativo alle informazioni di notifica dai Raspberry al server, ed essendo tali

informazioni non critiche (utilizzate per la generazione di statistiche), è tollerabile qualche sporadica perdita o, nel caso di configurazioni cluster, lieve inconsistenza delle informazioni. La quantità esatta di informazioni perse accettabili è dipendente dalla variabilità delle configurazioni impostate sui Raspberry: nel caso di configurazioni molto statiche, si potrebbe accettare di perdere anche il 15% di richieste, poiché queste sarebbero ricostruibili dai valori precedenti e successivi; in caso di configurazioni maggiormente dinamiche, sarebbe più opportuno un tasso di perdite inferiore al 3%. Nell'utilizzo tipico comunque si prevede che il tasso medio di perdita rimanga sotto l'1%: qualora dovesse aumentare, questo sarebbe un indice o di gravi problemi di connettività di rete, o di sottodimensionamento del server.

1.2.5 Scalabilità

Il sistema deve essere in grado di far fronte all'elevato numero di richieste, direttamente proporzionali al numero di Raspberry nel sistema. I lampioni da gestire sono nell'ordine di circa qualche migliaio, dipendentemente dall'ampiezza e dalla topologia della città che si intende coprire. Poiché il collo di bottiglia del sistema sarà costituito dai server sui quali vengono centralizzate le informazioni provenienti dai Raspberry, è necessaria un'infrastruttura in grado di far fronte alle richieste. L'implementazione tramite server cloud di tipo "Infrastructure as a Service" (IaaS) consente un rapido ridimensionamento delle risorse hardware allocate ai server, per far fronte alle richieste crescenti nel tempo (ipotizzando che i Raspberry vengano installati in maniera progressiva). Inoltre, in base al carico sui server, è possibile installare i vari servizi su un unico server oppure su molteplici, fino ad avere un server dedicato per ogni servizio. Infine, qualora le richieste risultassero ancora maggiori, si potrà configurare un cluster di server con load balancer, e multiple istanze dei servizi, oltre a nodi intermedi che fungano da intermediari tra i Raspberry e i server centrali. Ai fini di garantire la scalabilità, è necessario anche stabilire attentamente le modalità con cui archiviare i dati. Poiché la maggior parte dei dati archiviati sarà costituita dai messaggi di stato provenienti dai Raspberry, la quantità di tali messaggi sarà notevole, e l'elemento caratterizzante di ogni record è il timestamp, appare sensato utilizzare un "time series database" piuttosto che un database relazionale tradizionale. L'accesso ai dati infatti avverrà sulla base del tempo al quale saranno stati registrati. Inoltre, un database di questo tipo

rende agevole l'eliminazione o l'archiviazione offline dei dati più vecchi, al fine di liberare spazio e migliorare le prestazioni.

1.2.6 Robustezza

Il sistema dovrà essere tollerante ai guasti: il malfunzionamento di un Raspberry non deve compromettere il resto del sistema, ma solo le risorse direttamente collegate ad esso. Per quanto riguarda il server, essendo un "single point of failure", si può prevenire guasti configurando server ridondanti. In ogni caso, qualora il sistema server andasse del tutto offline o si verificassero problemi di rete che impediscono la navigazione, i Raspberry continuerebbero a funzionare localmente (rilevamento del passaggio di auto e accensioni e spegnimenti secondo l'ultima pianificazione impostata), garantendo ai tecnici tutto il tempo necessario per ripristinare il guasto.

1.2.7 Qualità del servizio adeguata

Tutti i Raspberry eseguono un comportamento, visibile al pubblico, gestito localmente. Di conseguenza i fruitori del servizio, ovvero gli automobilisti di passaggio, non subiscono disagi da eventuali problemi di comunicazione con il server centrale. Invece, dal punto di vista del server si richiede che le sue risorse siano tali da gestire indicativamente una cinquantina di richieste al giorno da ogni lampione (valore dipendente dalla topologia e dal traffico della città). Per l'accensione e spegnimento programmato dei lampioni non sono presenti vincoli stringenti in termini temporali, visto che anche un ritardo di qualche secondo, o persino minuto, nell'accensione o spegnimento non comprometterebbe la sicurezza degli automobilisti. Questo comportamento è comunque gestito localmente dai singoli Raspberry, per cui non è condizionato da problemi di rete. Per quanto riguarda l'accensione dei lampioni in sequenza dopo il rilevamento di un veicolo, bisogna garantire un buon livello di tempestività per evitare il transito dei veicoli con i lampioni ancora in modalità di risparmio energetico. Un tempo massimo adeguato per questo scopo, valutando una velocità dei veicoli non superiore ad 80 km/h (verosimilmente inferiore, trattandosi di centri abitati), si può stimare nell'ordine di qualche centinaio di millisecondi. Essendo il rilevamento e la successiva accensione comportamenti locali, tale vincolo potrà essere ampiamente rispettato senza particolari accorgimenti.

1.2.8 Facilità di gestione

Gli utenti amministratori devono essere in grado di modificare la configurazione dei Raspberry in maniera centralizzata, senza doversi recare fisicamente ad ogni singolo Raspberry. Potrà essere utile implementare più interfacce di gestione per scopi diversi. Le statistiche sono ugualmente consultabili in maniera centralizzata. Possono essere definiti utenti diversi per scopi diversi (consultazione statistiche, modifica configurazioni).

1.2.9 Sicurezza

Il sistema dovrà essere progettato e implementato utilizzando tecniche di “security by design”, quali una robusta crittografia nelle comunicazioni basata su certificati e autorità di certificazione e opzionalmente una VPN dedicata per non esporre i servizi direttamente sulla rete Internet. Solo gli utenti autorizzati dovranno essere in grado di modificare le configurazioni e accedere ai dati. Inoltre, la superficie di attacco dovrà essere ridotta al minimo e si dovrà mitigare l’impatto degli eventuali danni in caso di intrusione.

1.2.10 Adattabilità

Poiché gli scenari applicativi del sistema sono variabili, è importante garantire un buon grado di adattabilità per quanto riguarda le modalità con cui avvengono le comunicazioni di rete. Essendo la quantità di dati trasferita dai Raspberry molto contenuta, si aprono molteplici scenari. La gestione dei Raspberry può avvenire tramite rete cablata, Wifi (IEEE 802.11n) o 3G. La rete cablata è la scelta migliore per sicurezza, affidabilità, consumo energetico, resistenza ad interferenze elettromagnetiche e prestazioni (anche se in questo tipo di applicazione il volume di dati trasferiti è molto ridotto). Tuttavia, a causa della topologia della città e dall’eventuale mancanza di condotti sotterranei per il cablaggio, non è sempre possibile portare fisicamente un cavo Ethernet ad ogni Raspberry: in questi casi un metodo alternativo di connessione può essere il Wifi, nel caso si riesca a posizionare un access point nelle vicinanze, oppure il 3G, che consente di far comunicare i Raspberry senza bisogno di ulteriori dispositivi intermediari aggiuntivi, ma sfruttando le antenne già esistenti degli operatori. La percorribilità di queste ultime opzioni è dipendente dalla copertura di segnale e dalla possibile presenza di interferenze esterne.

1.3 Analisi del problema

TODO?

Capitolo 2

Progettazione

In questo capitolo viene descritta in maniera maggiormente dettagliata l'architettura del progetto e le scelte che hanno portato alla sua realizzazione.

2.1 Progettazione generale del sistema

Il progetto è stato diviso in due macro parti e in diverse sottoparti:

- Una parte in esecuzione su un sistema embedded, suddiviso in:
 - Rilevamento illuminazione ambientale con il sensore di luce
 - Rilevamento presenza di automobili con i sensori di prossimità
 - L'illuminazione di ogni singolo lampione in base alle politiche locali
 - Il programma principale che gestisce il web server REST e l'invio delle statistiche al server
- Una parte in esecuzione su un server che ??

Altre info qui.

2.2 Architettura complessiva

yes

2.3 Controllo presenza auto

Questa parte si occupa di controllare la presenza di eventuali auto in transito. Utilizzando un sensore di prossimità periodicamente viene calcolata la distanza tra il sensore e l'ostacolo più vicino per valutare se un'auto possa essere in transito in quel momento sulla carreggiata.

2.3.1 Comportamento

Per la rappresentazione dell'architettura è stato scelto di utilizzare una Finished State Machine.

Elenco delle figure