Appalto concorso per la realizzazione di High Performance & Highly Available Software HTTP Balancer As-A-Service

Proponente

New Vision è una software house Italiana che ha come mission quello di portare le aziende ad utilizzare Internet come principale mezzo di comunicazione.
Dal 2000 New Vision sviluppa le soluzioni con il team di ricerca e sviluppo basato nell'headquarter Italiano, ottenendo vari riconoscimenti internazionali per progetti innovativi.

Riferimenti per il capitolato

Coordinamento sviluppo: Fabio Tudone (fabio.tudone@newvision.it)
Responsabile e riferimento per il progetto "HP & HA software HTTP balancer as a service": Leonardo Scattola (leonardo.scattola@newvision.it)

Oggetto

Il profilo di carico di applicazioni web orientate alla comunicazione e all'interazione real-time può presentare notevoli variazioni temporali. Questo problema veniva normalmente gestito dimensionando l'infrastruttura per il *massimo carico ipotizzabile*, ma i progressi nell'ambito della virtualizzazione e il conseguente avvento del Cloud Computing di tipo *Infrastructure As A Service* consente la progettazione e l'utilizzo di infrastrutture elastiche, cioè che si possono ridimensionare rapidamente (e potenzialmente anche automaticamente) in base al profilo di carico del momento.
Un componente fondamentale delle infrastrutture scalabili utilizzate per web applications è il *balancer HTTP*, che si occupa di smistare le richieste HTTP in ingresso verso più di un server HTTP per ottenere maggiore affidabilità, maggiori prestazioni o entrambe le cose.
Un'infrastruttura elastica *as-a-service* richiede però un balancer HTTP con le stesse caratteristiche. Progetti open-source come *HAProxy* dimostrano inoltre che un balancer HTTP software è tipicamente molto più flessibile di un'appliance dedicata senza necessariamente essere meno performante.
Il progetto consiste nella progettazione e nella realizzazione di un (interfaccia per) balancer HTTP software con le caratteristiche necessarie per renderlo proficuamente utilizzabile nell'ambito di un'infrastruttura elastica di tipo cloud computing.

Requisiti

Funzionali Obbligatori

- Web Services ReSTful con aggiunta e rimozione di nodi dal gruppo, configurazione delle politiche di rimozione e ri-aggiunta automatica dei nodi (health check)
- Supporto a websocket
- Timeout delle connessioni HTTP configurabile per gruppo di bilanciamento e supporto a valori > 1 minuto
- Possibilità di utilizzo come libreria, oltre che come servizio HTTP

Non funzionali obbligatori

- Utilizzo di tecnologie basate su I/O asincrono dei moderni sistemi operativi, in modo che il numero di connessioni TCP attive in un dato momento (potenzialmente decine di migliaia) non sia vincolato dal numero di thread attivabili per un singolo processo (tipicamente al massimo qualche migliaio)
- Compatibillità con sistemi operativi Linux
- Basso footprint di memoria (sotto i 50 MB con 30.000 connessioni attive)
- Tempo di riavvio basso (< 1 sec)
- Riconfigurabilità a caldo senza perdita di connessioni
- Alta affidabilità del singolo nodo sotto carico (>= 99,99999% del tempo)
- Supporto ad almeno 30.000 connessioni contemporanee
- Codice ben strutturato e mantenibile
- Fornitura di AMI (*Amazon Machine Image*) Ubuntu Linux pronta all'utilizzo in AWS EC2 (http://aws.amazon.com/ec2/)

Funzionali opzionali

- Versione XML/SOAP dei Web Services
- Supporto a gruppi di bilanciamento distinti scelti sulla base di regole flessibili applicabili a tutte le informazioni (parametri, URL, hostname, headers, …) presenti nella richiesta HTTP in ingresso
- Interfaccia web di monitoring e gestione gruppi di bilanciamento e nodi registrati

Non funzionali opzionali

- Compatibilità delle API con aXApi di A10 Networks (http://www.a10networks.com/); i metodi interessati per la gestione dei server bilanciati dal balancer sono: `authenticate`, `session.close`, `slb.service-group.getByName`, `slb.service-group.update`
- Compatibilità con sistemi Windows rispettando gli stessi requisiti
- Funzionamento in cluster (configurazione allineata tra i nodi), bilanciabile ad es. da DNS o da ulteriore front balancer
- Monitoring del cluster tramite l'interfaccia web

Variazioni ai requisiti

Non sono ammesse variazioni ai requisiti se non a concordato miglioramento di quanto richiesto dal committente. E' possibile la comunicazione, da parte del committente, di variazioni ai requisiti per precisazione, chiarimento, integrazione, sia precedentemente alla consegna delle offerte che durante la realizzazione del sistema.

Obiettivi

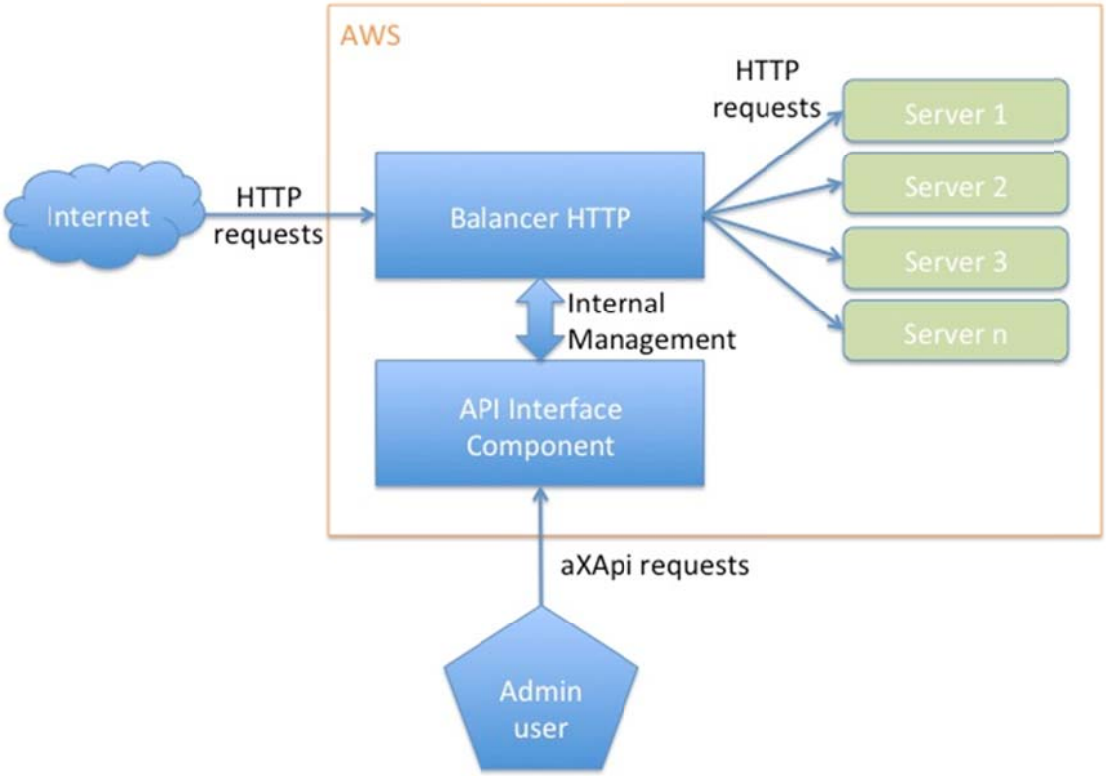
Il soddisfacimento parziale o totale dei requisiti non obbligatori è facoltativo, e costituisce titolo per la valutazione dell'offerta ai fini della vincita dell'appalto.
Il supporto di aXApi costituisce importante elemento sulla valutazione del progetto in quanto riduce l'impatto per l'adozione da parte del committente.
L'introduzione di nuove funzionalità non elencate nel presente capitolato verrà valutato positivamente solo nel caso in cui l'impatto negativo sugli altri parametri obbligatori sia nullo o trascurabile.
Il minor utilizzo di risorse hardware verrà valutato positivamente per la scelta del fornitore.
Il minor costo di licenza del sistema operativo sarà oggetto di valutazione per la scelta del fornitore.

Tecnologie suggerite

Tra le piattaforme che potrebbero soddisfare almeno i requisiti obbligatori espressi e consentire di risparmiare tempo nella realizzazione si citano:

- Amazon Web Services
- NGINX: HTTP server modulare con load balancing basato su asynchronous I/O e completamente *scriptable* in LUA (www.lua.org)
- HAProxy: load balancer software ad alte prestazioni basato su asynchronous I/O
- Node.JS (V8) con il modulo node_proxy ed eventuali altre librerie specializzate per il load balancing
- Linguaggi JVM (e.g. Java, Scala, Clojure, Groovy) usando librerie HTTP basate su Netty (ma con footprint molto, forse troppo, alto)
- Erlang
- Go
- C/C++

Ipotesi di architettura



Le componenti Balancer HTTP e API Interface Component sono oggetto del capitolato.
Il modulo Balancer HTTP può essere una delle tecnologie suggerite (ngnix, haproxy…), nel qual caso si richiede necessario lo sviluppo di un modulo di interfacciamento API REST in quanto la soluzione non fornisce la possibilità di essere gestita tramite API.

Documentazione

La consegna del sistema dovrà essere accompagnata dai necessari manuali di riferimento delle librerie e dei web services e delle relative guide utente (tutorial), dai codici sorgenti con licenza scelta dal fornitore, dai diagrammi architetturali e dalla documentazione per l'installazione, l'uso e la manutenzione della soluzione.

Garanzia e manutenzione

Il fornitore dovrà garantire in sede di collaudo il corretto funzionamento del sistema rispetto ai requisiti. L'eliminazione dei difetti e delle non conformità eventualmente emersi in sede di collaudo sono a totale carico del fornitore. Le modalità di collaudo saranno proposte dal fornitore e costituiranno titolo per la valutazione dell'offerta ai fini dell'aggiudicazione dell'appalto. I dati di collaudo costituiscono parte integrante delle modalità di collaudo. Le modalità di collaudo saranno considerate definitive e contrattualmente vincolanti solo a seguito di formale approvazione da parte del committente.

Rinvio

Per tutto quanto non previsto nel presente capitolato sono applicabili le disposizioni contenute nelle leggi e nei collegati per la gestione degli appalti pubblici.

Risorse utili

- Documentazione nginx: http://wiki.nginx.org/Main
- Documentazione HAproxy: http://code.google.com/p/haproxy-docs/
- Documentazione Node.JS (V8): http://nodejs.org/api/index.html
- A10 Networks aXApi documentation: http://digitalfreaks.org/~lavalamp/AX_Documentation_CD_v2_4_3-20100621/AX_Documentation_CD_v2_4_3-20100621/AX_aXAPI_Ref_v2_4_3-20100621.pdf
- Varnish https://www.varnish-cache.org/trac/wiki/LoadBalancing
- Cherokee http://www.cherokee-project.com/doc/modules_balancers.html
- Perlbal (anche se il fatto che sia in Perl non mi entusiasma): https://github.com/perlbal/Perlbal
- Limiti load-balancing in AWS: http://agiletesting.blogspot.it/2010/04/load-balancing-in-cloud-whitepaper-by.html