
Ausarbeitung

Loadbalancing

DEZSYS - Dezentrale Systeme
5YHITM 2016/17

Maximilian Seidl

Note:
Betreuer: Th.Micheler, M.Schabel

Version 0.1
Begonnen am 18. Oktober 2016
Beendet am 18. Oktober 2016

Inhaltsverzeichnis

1	Was ist Load Balancing?	1
2	Serverlastverteilung	1
3	Verfahren von SLB	1
3.1	DNS Round Robin	2
3.1.1	Aufbau und Funktionsweise	2
3.2	NAT based SLB	2
3.3	Flat based SLB	2
3.4	Anycast SLB	2
3.5	Tabelle	2
3.6	Aufzählung	2
3.7	Code	3

1 Was ist Load Balancing?

Der Begriff Load Balancing (zu deutsch Lastverteilung) beschreibt umfangreiche Berechnungen oder große Mengen von Anfragen, welche auf mehrere parallel arbeitende System verteilt werden. Es gibt verschiedene Ausprägungen, welche mit der Menge* der Kommunikation skalieren. Diese reichen von Lastverteilung auf einem System mit mehreren Prozessoren, bis hin zu ganzen Computer-Clustern, welche die Arbeit auf mehrere Rechner aufteilen.

2 Serverlastverteilung

Serverlastverteilung (englisch Server Load Balancing "SLB") kommt überall dort zum Einsatz, wo sehr viele Clients eine hohe Anfragedichte erzeugen und damit einen einzelnen Server-Rechner überlasten würden.

Typische Kriterien zur Ermittlung der Notwendigkeit von SLB sind die **Datenrate**, die **Anzahl der Clients** und die **Anfragerate**.

Die Erhöhung der Datenverfügbarkeit wird durch SLB gefördert. Redundante Datenhaltung wird durch das Einsetzen von mehreren Systemen ermöglicht. Die Aufgabe des SLB ist hier die Vermittlung der Clients an die einzelnen Server.

3 Verfahren von SLB

Es bieten sich einige verschiedene Verfahren, welche bei Server Load Balancing zum Einsatz kommen können. Ein mögliches Verfahren wäre das Vorschalten eines Systems (**Load Balancer, Frontend Server**), der Anfragen aufteilt, oder die Verwendung von DNS (Domain-Name-System) mit dem sogenannten **Round-Robin-Verfahren**. Bei Webservern darf eine Serverlastverteilung nicht fehlen, da ein einzelner Host nur eine begrenzte Anzahl an HTTP-Anfragen auf einmal beantworten kann. Es besteht auch die Möglichkeit, dass der Load Balancer die Verschlüsselung zum Client übernimmt und intern die Anfragen auf seine Art und Weise verarbeitet.

Es bestehen folgende Verfahren:

- DNS Round Robin
- NAT based SLB
- Flat based SLB
- Anycast SLB

3.1 DNS Round Robin

Lastverteilung per DNS (englisch Round robin DNS) ist die einfachste Technik zur SLB auf Basis des DNS. Da es aber zu einem Caching der DNS-Antworten kommt, ist dieses Verfahren nur manchmal wirklich sinnvoll.

3.1.1 Aufbau und Funktionsweise

DNS lässt es zu, dass einem Namen mehrere IP-Adressen zugewiesen werden können. Dies bedeutet es können mehrere sogenannte **Resource Records** mit gleichem Label, gleicher Klasse und gleichem Typ entstehen. Eine derartige Anordnung wird als *Resource Record Set* bezeichnet.

server.example.com.	1800	IN	A	192.0.2.70
server.example.com.	1800	IN	A	192.0.2.71
server.example.com.	1800	IN	A	192.0.2.72

Listing 1: DNS Round Robin Example

Wird ein derartiger Name von einem Resolver abgefragt, so liefert der DNS-Server grundsätzlich alle bekannten IP-Adressen zurück, allerdings in wechselnder Reihenfolge. Der erste Request wird dann beispielsweise mit [192.0.2.70, 192.0.2.71, 192.0.2.72] beantwortet und der zweite mit [192.0.2.71, 192.0.2.72, 192.0.2.70]. Es liegt dann in der Verantwortung des Resolvers, welche IP-Adresse er tatsächlich verwendet.

Nach welcher Strategie ein DNS-Server die Reihenfolge vorgibt, kann bei Bind-kompatiblen Nameservern konfiguriert werden. Bei BIND sind drei Varianten möglich: zyklisch, zufällig und fest. Bei der Variante fest werden die IP-Adressen in der Reihenfolge zurückgegeben, in der sie im Nameserver abliegen.

Anmerkung: Bei reversen Zonen der IN-ADDR.ARPA-Domäne ist ein Loadbalancing nicht möglich, obwohl für eine IP-Adresse mehrere Namen definiert werden können. Eine Lastverteilung wäre hier auch nicht sinnvoll.

3.1.2 Technischer Fortschritt

Bei moderneren Resource-Record-Typen wie SRV oder NAPTR lässt sich außerdem noch eine Gewichtung definieren, die festlegt, welche Server-IP-Adressen am häufigsten an erster Stelle stehen. Die entsprechenden Server werden damit häufiger angesprochen.

Bei Record-Typen, die keine Gewichtungparameter zur Verfügung stellen, besteht die etwas unschöne, aber machbare Alternative darin, die IP-Adressen entsprechend ihrer Gewichtung mehrfach zu vergeben, z. B. ADSL-Leitung dreimal, Funkstrecke nur einmal.

Außerdem gibt es die Möglichkeit, aus einem Pool von möglichen Servern nur einige zurückzuliefern. So werden beispielsweise vom Google-Nameserver immer drei IP-Adressen zurückgeliefert, die teilweise wechseln. Sinnvoll ist auch eine standortbezogene Rücklieferung von IP-Adressen, wenn mehrere verteilte Rechenzentren zur Verfügung stehen – dies wird z. B. von CDNs genutzt.

3.1.3 Nachteile

Die Lastverteilung durch DNS ist natürlich nur in dem Sinn gleichmäßig, was die Zuteilung betrifft. Über die danach entstehende tatsächliche Belastung weiß DNS nichts. Auch wird nicht überprüft, ob die Zielservers überhaupt ansprechbar sind. Vorgeschaltete Skripts können aber die Verfügbarkeit prüfen und nur diejenigen Server im Nameserver eintragen, die aktuell tatsächlich zur Verfügung stehen. Damit lassen sich Lastverteilung und Ausfallsicherung verbinden.

3.2 NAT based SLB

3.3 Flat based SLB

3.4 Anycast SLB

3.5 Tabelle

Header	Kopf
Lorem	Ipsum dolor sit amet, consetetur sadipscing elitr
Ipsum	At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus
Dolor	Consetetur sadipscing elitr, sed diam nonumy

Tabelle 1: Lorem ipsum dolor sit amet [1]

3.6 Aufzählung

- **Lorem ipsum:** dolor sit amet, consetetur sadipscing elitr
- sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat
- ut labore et dolore magna aliquyam erat, sed diam voluptua

3.7 Code

At vero eos et accusam et justo duo dolores et ea rebum.

```
try{
  gTransCur.begin();
  //Perform the operation inside the transaction
  not_registered =
    gRegistrarObjRef.register_for_courses(student_id,selected_course_numbers);
}
```

```
    if (not_registered != null)

        //If operation executes with no errors, commit the transaction
        boolean report_heuristics = true;
        gTransCur.commit(report_heuristics);

    } else gTransCur.rollback();

} catch(org.omg.CosTransactions.NoTransaction nte) {
    System.err.println("NoTransaction: " + nte);
    System.exit(1);
} catch(org.omg.CosTransactions.SubtransactionsUnavailable e) {
    System.err.println("Subtransactions Unavailable: " + e);
    System.exit(1);
} catch(org.omg.CosTransactions.HeuristicHazard e) {
    System.err.println("HeuristicHazard: " + e);
    System.exit(1);
} catch(org.omg.CosTransactions.HeuristicMixed e) {
    System.err.println("HeuristicMixed: " + e);
    System.exit(1);
}
```

Listing 2: Implizite Transaktion [1]

Literatur

- [1] A.S. Tanenbaum and M. Van Steen. *Verteilte Systeme: Prinzipien und Paradigmen*. Pearson Studium. Addison Wesley Verlag, 2007.

Tabellenverzeichnis

1	Lorem ipsum dolor sit amet [1]	3
---	--	---

Listings

1	DNS Round Robin Example	2
2	Implizite Transaktion [1]	3

Abbildungsverzeichnis