Lastbalancierung mit Random Walks

Effiziente Algorithmen – Sommer 2019

Martin Hoefer

Lastbalancierung im Netzwerk mit Thresholds:

- Gegeben: Ungerichteter Graph $G = (V, E)$
- Der Einfachheit halber: G nicht bipartit (Resultate gelten auch für bipartit)
- Kanten sind Direktverbindungen (Strassen, Kabel, etc.)
- Jeder Knoten ist hat einen Speicher (Rechnerspeicher, Parkplatz für Autos, etc.)

- Es gibt m **Lasteinheiten** (Tasks, Mietautos, E-Roller, etc.),
- Anfangs sind die Lasteinheiten beliebig im Netzwerk verteilt
- Jeder Knoten v hat **Anzahl freier Plätze** $\tau_v \in \{1, 2, \ldots, m\}$ (**threshold**)
- Wir nehmen an:
$$\sum_{v \in V} \tau_v \geq m.$$
- Verteiltes Netzwerk, Knoten verschieben überschüssige Lasteinheiten zu ihren Nachbarn

Ziel: Balancierte Aufteilung der m Einheiten, so dass kein Knoten mehr als τ_v Einheiten hat

---

Algorithm 1: Verteiltes Random-Walk Protokoll

Alle Lasteinheiten sind aktiv
for *Runde* $t = 1, 2, 3, \ldots$ *an jedem Knoten* v *parallel* **do**
    Sei ℓ_v^t die Last von Knoten v in Runde t
    if $(\ell_v^t \leq \tau_v)$ **then**
        Alle Lasteinheiten auf v werden passiv.
    else
        v wählt $\ell_v^t - \tau_v$ aktive Lasteinheiten beliebig aus
        Alle anderen Einheiten auf v werden passiv
        for *jede aktive Lasteinheit bei* v **do**
            Wähle uniform zufällig einen Nachbarn von v aus
            Schicke dem Nachbarn die Lasteinheit

Wo ist hier ein Random Walk?

- Anfangs sind alle m Lasteinheiten **aktiv**. Jeder Knoten v hat τ_v freie Plätze.
- Wenn eine Lasteinheit auf ihrem momentanem Knoten einen freien Platz ergattert, bleibt sie dort liegen und wird **passiv**.
- Ansonsten wird sie vom Knoten zu einem zufälligen Nachbarn weitergeschickt
- Alle **aktiven Lasteinheiten** machen gleichzeitig **parallele Random Walks**

Wieviele Runden bis eine balancierte Aufteilung erreicht ist? Wann sind alle Einheiten passiv?

Für eine Aufteilung ℓ der Lasteinheiten sei

$$\Phi(\ell) = \sum_{v \in V} \max(\ell_v - \tau_v, 0)$$

die **Anzahl der aktiven Einheiten** in ℓ.

Theorem 1 (H., Sauerwald, 2013 [1])**.** *Sei $H(G) = \max_{u,v \in V} H(u,v)$ die maximale Hitting-Time eines Knotens in G. Sei ℓ^0 die Startaufteilung der Lasteinheiten. Dann erreicht das Random-Walk Protokoll eine balancierte Aufteilung nach erwarteter Rundenanzahl von*

$$O(H(G) \cdot \log \Phi(\ell^0)).$$

Das Theorem folgt aus folgenden Lemma:

Lemma 1. *Sei ℓ^0 eine beliebige Lastaufteilung der Einheiten. Betrachte die erste Runde T mit $\Phi(\ell^T) \leq \frac{7}{8} \cdot \Phi(\ell^0)$. Dann ist $\mathbb{E}[T] = O(H(G))$.*

Beweis des Lemmas:
Anfangs $\Phi(\ell^0)$ aktive Einheiten und (mindestens) genauso viele freie Plätze im Netzwerk.

- Ordne jeder aktiven Einheit genau einen freien Platz auf einem der Knoten zu.
- Jeder freie Platz wird höchstens einmal vergeben.
- Wieviele Einheiten erreichen ihre zugewiesenen Plätze in welcher Zeit?

Sei x aktive Einheit, anfangs auf Knoten u, die auf den freien Platz bei Knoten v soll. Wie lange braucht x bis sie bei v ankommt?

- Wir **nehmen erstmal an**, jede Einheit x bleibt **für immer aktiv**.
- Sei $T(x)$ die Anzahl Runden bis x Knoten v erreicht, dann ist $\mathbb{E}[T(x)] = H(u,v) \leq H(G)$.
- Mit Markoff-Ungleichung:
$$\Pr[T(x) \geq 2H(G)] \quad \leq \quad \frac{1}{2} \; .$$

Wieviele Einheiten haben nach $2H(G)$ Runden ihren Zielknoten **mindestens einmal besucht**?

- Sei $R(x) = 1$ wenn x ihren Zielknoten in den $2H(G)$ Runden besucht hat und 0 sonst.
- Jede Einheit x macht **unabhängigen Random Walk** und hat **Wahrscheinlichkeit mindestens 1/2** ihren Zielknoten mindestens einmal zu besuchen.
- Also
$$\mathbb{E}\left[\sum_{x \text{ aktiv}} R(x)\right] \quad \geq \quad \frac{\Phi(\ell^0)}{2}$$
und mit Chernoff-Schranke
$$\Pr\left[\sum_{x \text{ aktiv}} R(x) \leq \frac{1}{2} \cdot \frac{\Phi(\ell^0)}{2}\right] \leq e^{-\Phi(\ell^0)/8} \leq e^{-1/8} \quad ,$$
da $\Phi(\ell^0) \geq 1$ (sonst wären wir schon am Anfang balanciert).

Wie lange dauert es, **bis mindestens $\Phi(\ell^0)/4$ Einheiten** ihren jeweiligen Zielknoten **mindestens einmal besucht** haben?

- Sei K das Ereignis, dass das innerhalb der ersten $2H(G)$ Runden klappt.
- K tritt ein wenn $\sum_{x \text{ aktiv}} R(x) > \Phi(\ell^0)/4$, also
$$\Pr[K] \geq 1 - e^{-1/8} \enspace .$$
- Wenn K nicht eintritt, dann nehmen wir an, der Prozess startet neu.
- Die Wahrscheinlichkeit, dass es nach k Neustarts immer noch nicht geklappt hat, ist $(e^{-1/8})^k$.
- Also hat es nach erwartet höchstens
$$\begin{aligned} &\sum_{k=0}^{\infty} (e^{-1/8})^k \cdot (1 - e^{-1/8}) \cdot (k+1) \cdot 2H(G) \\ &= (1 - e^{-1/8}) \cdot 2H(G) \cdot \sum_{k=0}^{\infty} (e^{-1/8})^k \cdot (k+1) \\ &= \frac{2}{1 - e^{-1/8}} \cdot H(G) \end{aligned}$$
vielen Runden geklappt.

Nach **erwartet $O(H(G))$ vielen Runden** haben mindestens $\Phi(\ell^0)/4$ **Einheiten** ihren jeweiligen **Zielknoten mindestens einmal** besucht.

Wir nehmen bisher noch an, dass alle **aktiven Einheiten ewig aktiv** sind. Sei M die **Menge der ersten $\Phi(\ell^0)/4$ Einheiten**, die ihren Zielknoten besuchen (wenn sie ewig aktiv sind).
- Einheiten in M sollten **alle passiv werden** und den Random Walk abbrechen sobald sie ihren jeweiligen Zielknoten erreichen.
- Protokoll weiss aber nicht, welche Einheit auf welchen Platz soll. Was kann schiefgehen?

Problem 1: x bekommt Platz auf anderem Knoten, bricht Random Walk ab bevor sie v besucht.
Problem 2: x besucht v, aber jemand anders hat v den Platz "geklaut".

Problem 1 ist gut für v, wird aber evtl. Problem 2 für eine andere Einheit. Also Problem 2:
- Platz ist geklaut, statt x ist eine andere aktive Einheit nun passiv.
- Wenn also $x \in M$ ewig aktiv wäre, auf ihrem Knoten ankäme und dort merken würde, dass ihr Platz geklaut wurde, dann ist vorher **genau eine eindeutige andere aktive Einheit** schon passiv geworden.
- Im worst-case ist diese andere Einheit auch aus M.

Es sind **evtl. nicht alle** aber doch insgesamt **mindestens $M/2 = \frac{1}{8} \cdot \Phi(\ell^0)$ Einheiten passiv** geworden. Mit der Definition von M folgt, dass dies nach $O(H(G))$ Runden passiert. □

Beweis des Theorems:
Wir teilen den Prozess nun in Phasen ein.
- Phase 1 startet in Runde 1. Phase $i = 1, 2, 3, \ldots$ besteht aus den Runden t mit
$$(7/8)^{i-1} \cdot \Phi(\ell^0) \quad \geq \quad \Phi(\ell^t) \quad > \quad (7/8)^i \cdot \Phi(\ell^0)$$
aktiven Einheiten.
- Wende Lemma 1 für jede Phase i an: Die erwartete Länge von Phase i ist $O(H(G))$.
- Es gilt $\Phi(\ell^t) \in \{0, 1, 2, \ldots\}$. Wenn $(7/8)^{i^*-1} \cdot \Phi(\ell^0) \geq 1 > (7/8)^{i^*} \cdot \Phi(\ell^0)$, dann muss i^* die letzte Phase sein.
- i^* ist das kleinste i mit $(7/8)^i \cdot \Phi(\ell^0) < 1$. Es gilt $i^* = \lceil \log_{8/7} \Phi(\ell^0) \rceil$.

Es gibt also nur $O(\log \Phi(\ell^0))$ Phasen, jede Phase hat erwartete Länge $O(H(G))$. □

Literatur

[1] M. Hoefer, T. Sauerwald. Threshold Load Balancing in Networks. *CoRR* abs/1306.1402, 2013. Brief Announcement in PODC 2013.