

PS Algorithmen für verteilte Systeme

EINGEREICHT VON

BAUMGARTNER DOMINIK, DAFIR SAMY

GRUPPE 1(16:00)

Aufgabe 11:

Schreiben Sie ein Programm, das den Push-Algorithmus auf einem vollständigen Graphen der Größe 10.000 simuliert und geben Sie die Verteilung der Laufzeiten von 100 unterschiedlichen Simulationsdurchläufen aus.

Aufgabe 12:

Zeigen Sie, dass der Push-Algorithmus im vollständigen Graphen der Größe $n > 10^6$ mit positiver Wahrscheinlichkeit eine Laufzeit von mindestens $\log_2(\frac{n}{2}) + \frac{\log_4(n)}{4}$ besitzt. Beachten Sie, dass für die Chernoff-Schranke folgende Monotonie-Eigenschaft gilt:

$$\Pr[X \leq (1 - \delta)\mu'] \leq e^{-\frac{\delta^2 \mu'}{3}} \text{ für alle } \mu' \leq \mu.$$

- Zuerst wird gezeigt, dass mit Wahrscheinlichkeit $(\frac{1}{n})^{ld(\frac{n}{2}) + \frac{\log_4(n)}{4}}$ Knoten v_0 für $ld(\frac{n}{2}) + \frac{\log_4(n)}{4}$ Schritte ausschließlich mit sich selbst kommuniziert.

Zu Beginn ist nur v_0 informiert. Die Wahrscheinlichkeit, dass dieser Knoten nun in einem Schritt sich selbst wählt ist $\frac{1}{n}$. Somit ist Die Wahrscheinlichkeit, dass der Knoten sich in $ld(\frac{n}{2}) + \frac{\log_4(n)}{4}$ Schritten ausschließlich selbst wählt gleich $(\frac{1}{n})^{ld(\frac{n}{2}) + \frac{\log_4(n)}{4}}$.
jo

- Zeigen nun, dass man mindestens $ld(\frac{n}{2})$ Schritte benötigt, um $\frac{n}{2}$ Knoten zu informieren.

Wissen, dass in jedem Schritt die Anzahl der informierten Knoten sich verdoppelt. Also: $2^0 \rightarrow 2^1 \rightarrow 2^2 \rightarrow \dots \rightarrow 2^x$

Nach x Schritten sollen mindestens $\frac{n}{2}$ Knoten informiert sein:

$$2^x = \frac{n}{2}$$

$$\Rightarrow x = ld(\frac{n}{2})$$

→ Besitzt eine Laufzeit von $ld(\frac{n}{2})$

- Sei $t_0 = ld(\frac{n}{2})$ und definiere $\{v_1 \dots v_{\sqrt{n}}\} = \tilde{H}(t_0) \subset H(t_0)$ bzw. $\tilde{H}(t) = H(t) \cap \tilde{H}(t_0)$. O.B.d.A. seien $v_1 \dots v_k \in \tilde{H}(t)$ für $k = |\tilde{H}(t)|$, wobei $t > t_0$. Teste der Reihe nach Knoten $v_1 \dots v_k$, ob v_i von einem Knoten des Graphen kontaktiert wird. Es ist zu zeigen, dass Knoten v_i mit Wahrscheinlichkeit

$$\geq (1 - \frac{1}{n-i+1})^n \geq (1 - \frac{1}{n-k})^n$$

von keinem Knoten kontaktiert wird und diese Schranke unabhängig von allen anderen Knoten aus $\tilde{H}(t)$ gilt.

Gehen vom Best-Case aus, dass kein informierter Knoten, einen bereits von einem anderen informierten Knoten gewählten uninformatierter Knoten auswählt. Dass zu Beginn ein Knoten nun unseren Knoten v_i auswählt ist $\frac{1}{n}$. Pro Knoten den wir nun Durchlaufen, gibt es immer einen Knoten weniger zur Auswahl, da ja bereits vorher ein anderer Knoten informiert worden ist und dieser nicht noch einmal ausgewählt wird. \Rightarrow Je mehr Knoten nun in $\tilde{H}(t)$ durchlaufen wurden, umso weniger Knoten kann ein informierter auswählen, d.h. $\frac{1}{n-(i-1)}$ ist die Wahrscheinlichkeit, dass unser Knoten ausgewählt wird. Somit ist die Wahrscheinlichkeit, dass unser Knoten von keinem Knoten ausgewählt wird gleich $1 - \frac{1}{n-(i-1)}$. Da nun aber nicht nur ein Knoten auswählen kann sonder n verschiedene gilt: $(1 - \frac{1}{n-(i-1)})^n$. Wenn nun $i = k$, dann folgt, dass $\frac{1}{n-(k-1)} > \frac{1}{n-k}$ und somit gilt, dass Knoten v_i mit Wahrscheinlichkeit $(1 - \frac{1}{n-i+1})^n \geq (1 - \frac{1}{n-k})^n$ von keinem Knoten ausgewählt wird. Da wir alles global betrachtet haben und in jedem Schritt immer ein Knoten weniger wird, welcher aber nicht in $\tilde{H}(t)$ liegen muss, ist diese Schranke unabhängig von $\tilde{H}(t)$.

- Aus $(1 - \frac{1}{n-i+1})^{n-k} = \frac{1}{e}$ folgt $(1 - \frac{1}{n-i+1})^n > \frac{1}{3}$. Zeigen nun, dass mit einer „guten“ Wahrscheinlichkeit $|\tilde{H}(t+1)| \geq \frac{|\tilde{H}(t)|}{4}$, solange $|\tilde{H}(t)| \geq \sqrt[4]{n}$.

Wenn nun $(1 - \frac{1}{n-i+1})^{n-k} = \frac{1}{e}$ folgt, dass nun nach n Schritten, welche mehr sind als $n - k$ der Ausdruck $(1 - \frac{1}{n-i+1})^n$ größer werden muss. Da nun $e < 3$ und somit $\frac{1}{e} > \frac{1}{3}$ folgt $(1 - \frac{1}{n-i+1})^n > \frac{1}{3}$.

Annahme $|\tilde{H}(t)| \geq \sqrt[4]{n}$.

Sei $x_i = \begin{cases} 1 & \text{falls } v_i \text{ uninformatiert bleibt} \\ 0 & \text{sonst} \end{cases}$

$$X = \sum_{i=1}^{|\tilde{H}(t)|} x_i$$

$$\Rightarrow E(X) = E(\sum_{i=1}^{|\tilde{H}(t)|} x_i) = \sum_{i=1}^{|\tilde{H}(t)|} E(x_i) = \frac{|\tilde{H}(t)|}{3}$$

Verwenden nun Chernoff-Schranke mit $\mu' = \mu$:

$$P(X \leq (1 - \delta)\mu) = P(X \geq (1 + \delta)\frac{\sqrt[4]{n}}{3}) \leq e^{\frac{-\delta^2 \mu}{3}} = e^{\frac{-\delta^2 \sqrt[4]{n}}{3}}$$

Berechnen nun δ :

$$(1 - \delta)\mu = \frac{|\tilde{H}(t)|}{4}$$

$$\Rightarrow \delta = 1 - \frac{\frac{|\tilde{H}(t)|}{4}}{\frac{|\tilde{H}(t)|}{3}} = \frac{1}{4}$$

Somit ist

$$P(X \leq (1 - \delta)\mu) \leq \frac{1}{\sqrt[4]{144}}$$

Sei nun $n = 10^6$, dann folgt:

$$P(X \leq (1 - \delta)\mu) \leq \frac{1}{\sqrt[4]{144}} = 0,8028$$

Damit erhalten wir also eine kleine, aber positive Wahrscheinlichkeit dafür, dass $\frac{1}{4}$ der Ausgangsknoten uninformiert bleiben.

Nun stellt sich die Frage wie viele Schritte benötigt werden, um von \sqrt{n} Knoten zu den übrig bleibenden $\sqrt[4]{n}$ Knoten zu gelangen. Da gilt: $\sqrt{n} = \sqrt[4]{n} * \sqrt[4]{n}$ und $\sqrt[4]{n}$ Knoten übrig bleiben, verwenden wir nun den $ld(\sqrt[4]{n})$ (wieder eine best case Abschätzung), um die nötigen Schritte bei Halbierung in jedem Schritt zu erhalten. Durch umformen erhalten wir $ld(\sqrt[4]{n}) = \frac{1}{4} * ld(n)$ als benötigte Anzahl an Schritten um $\sqrt[4]{n}$ übrige Knoten zu erhalten. Je größer die Basis desto kleiner das Ergebnis einer Logarithmus. Deshalb legen wir $\frac{\log_4(n)}{4}$ als untere Schranke für diesen Ausdruck fest. Damit ist gezeigt, dass dieser Teil mindestens $\frac{\log_4(n)}{4}$ Schritte braucht und, dass dann immer noch $\sqrt[4]{n}$ Knoten übrig bleiben.

Es wurde gezeigt, dass der Push Algorithmus im vollständigen Graphen der Größe $n > 10^6$ sowohl im Sonderfall, in dem der erste informierte Knoten nur mit sich selbst kommuniziert, wie auch im regulären Fall mit positiver Wahrscheinlichkeit eine Laufzeit von mindestens $\log_2(\frac{n}{2}) + \frac{\log_4(n)}{4}$ hat. Weiters wurden hier immer Extremfälle betrachtet und nur eine echte Teilmenge der uninformierten Knoten verwendet. Damit ist das gezeigte auch für alle Normalfälle gültig. Somit besitzt der Push-Algorithmus im vollständigen Graphen der Größe $n > 10^6$ mit positiver Wahrscheinlichkeit eine Laufzeit von mindestens $\log_2(\frac{n}{2}) + \frac{\log_4(n)}{4}$