Распределённые алгоритмы

mk.cs.msu.ru → Лекционные курсы → Распределённые алгоритмы

Блок 45

Выявление конца вычислений:
алгоритм Дейкстры — Шолтена

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2025, февраль–май

Общее описание

Алгоритм Дейкстры-Шолтена — это обсуждающийся далее алгоритм ВКВ, предназначенный для

- произвольного связного неориентированного графа топологии $G = (V, E)$ и
- произвольного централизованного базового алгоритма:
  - один узел — инициатор,
  - остальные — последователи, и
  - последователь первым действием обязан принять сообщение

Контрольный алгоритм Дейкстры-Шолтена также централизован и имеет того же инициатора, что и базовый алгоритм

p^* — так дальше будем обозначать узел-инициатор

Общее описание

Контрольным алгоритмом строится и постоянно обновляется граф вычисления $T = (V_T, E_T)$:

1. V_T состоит из узлов и сообщений и включает в себя по крайней мере все активные узлы
2. Если $V_T \neq \emptyset$, то T — дерево, ориентированное к корню p^*

Контрольные действия узла p устроены так:

- Если p не входит в T и получил базовое сообщение m, то он добавляется в T, считая родителем отправителя m
- На все полученные базовые сообщения, кроме первого, p отправляет подтверждение (**ack**)
- Если p пассивен и все его отправленные базовые сообщения подтверждены, то он подтверждает оставшееся базовое сообщение и удаляется из T

Узел p^* выполняет $\mathrm{announce}$, когда удаляется из T

Код

Контрольные переменные узла p:

- $cou_p : \mathbb{N}_0/0$
- $parent_p : V^{\perp}$; начальное значение — если $p = p^*$, то p^*, а иначе $\perp$

$S_p(m, q)$: $cou_p := cou_p + 1$;

I_p: Если $cou_p = 0$, то $Leave_p$

$R_p(m, q)$:

1. Если $parent_p = \perp$:
   $parent_p := q$;
2. Иначе: $send_q(\mathbf{ack})$

Процедура A_p: $\{$есть сообщение **ack**, ещё не принятое узлом $p\}$

1. $receive(\underline{\mathbf{ack}})$
2. $cou_p := cou_p - 1$;
3. Если $cou_p = 0$ и $active_p = \mathbb{f}$: $Leave_p$

Процедура $Leave_p$ в составе I_p и A_p:

1. Если $parent_p = p$: $announce$
2. Иначе: $send_{parent_p}(\mathbf{ack})$
3. $parent_p := \perp$;

Корректность

Сообщение m, адресованное узлу p, далее будем обозначать записью $m_{\to p}$, а отправленное узлом p — $m_{p\to}$

Для заданной конфигурации γ дерево $T_\gamma = (V_\gamma, E_\gamma)$ зададим так

V_γ состоит из следующих вершин:

1. Каждый узел p, для которого $parent_p \neq \perp$
2. Каждое отправленное и ещё не принятое базовое сообщение (все такие сообщения считаются попарно различными)
3. Каждое отправленное и ещё не принятое подтверждение (все подтверждения считаются попарно различными)

E_γ состоит из следующих дуг:

1. $(p, parent_p)$ для каждого узла p, такого что $parent_p \notin \{\perp, p\}$
2. $(m_{p\to}, p)$ для каждого отправленного и ещё не принятого сообщения $m_{p\to}$
3. $(\mathbf{ack}_{\to p}, p)$ для каждого отправленного и ещё не принятого сообщения $\mathbf{ack}_{\to p}$

Корректность

Безопасность алгоритма Дейкстры-Шолтена обеспечивается инвариантом

$$P_{d-sch}(\gamma) = p_1 \,\&\, p_2 \,\&\, p_3 \,\&\, p_4 \,\&\, p_5,$$

где:

p_1: $\forall p \in V, active_p : p \in V_\gamma$

- То есть все активные узлы входят в T_γ

p_2: $\forall$ узлов и сообщений $u,\ v : (u, v) \in E_\gamma \Rightarrow u \in V_\gamma \,\&\, v \in V_\gamma \cap V$

- То есть T_γ — корректный граф, все дуги которого ведут в узлы

p_3: $\forall p \in V : cou_p = |\{v | \exists p : (v, p) \in E_\gamma\}|$

- То есть cou_p — это количество детей узла p в T_γ

p_4: $V_\gamma \neq \emptyset \Rightarrow T_\gamma$ — дерево с корнем p^*

p_5: $\forall p \in V : active_p = \mathbb{f} \,\&\, cou_p = 0 \Rightarrow p \notin V_\gamma$

- То есть пассивный узел не может быть листом дерева

Корректность

Лемма (Д.з. 1). P_{d-sch} **— инвариант алгоритма Дейкстры-Шолтена**

Теорема (Д.з. 2). **Алгоритм Дейкстры-Шолтена — это алгоритм ВКВ, в котором отправляется столько же контрольных сообщений, сколько и базовых**

Из теоремы 2 о сложности ВКВ следует, что алгоритм Дейкстры-Шолтена оптимален по числу отправляемых контрольных сообщений