Распределённые алгоритмы

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

Блок 34

Избрание лидера в дереве

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

ВМК МГУ, 2023/2024, весенний семестр

Если алгоритм избрания лидера* предназначен для топологии дерева (с хотя бы двумя узлами), то можно основать его на древесном волновом алгоритме

Но чтобы преобразовать древесный волновой алгоритм в такой алгоритм избрания лидера ($\mathcal{A}$), потребуется преодолеть два различия их свойств:

1. В волновом алгоритме инициаторами являются все листья дерева, а в алгоритме избрания лидера инициаторы должны быть произвольны, в том числе, быть может, ни одного листа
   - Для решения этой проблемы волна будет предварена фазой пробуждения узлов, в которой будут пересылаться сообщения типа **wakeup**, пробуждающие
2. В волновом алгоритме *как он был рассказан ранее* решение принимают ровно два узла, а в алгоритме избрания лидера следует в каждом узле выполнить либо leader, либо lost
   - Есть *расширенный вариант* древесного волнового алгоритма, в котором узел, принявший решение, отправляет фишки своим детям, заставляя и их принять решение
   - В $\mathcal{A}$ так будет рассылаться идентификатор лидера

Переменные узла:

- $s_p : bool = \mathbb{f}$
  - Отправлено ли пробуждающее сообщение
- $r_p : 2^{Neigh_p} = \emptyset$
  - От кого принято пробуждающее сообщение
- $X_p : 2^{Neigh_p} = Neigh_p$
  - От кого ещё не принято сообщение о лидере
- $v_p : \mathcal{T} = p$
  - $\mathcal{T}$ — тип идентификаторов узлов
  - Идентификатор узла, наиболее подходящего на роль лидера по текущему мнению p

Код узла p:

1. $Wake_p$ — фаза пробуждения
2. $Wave_p$ — фаза волны

Фаза пробуждения $Wake_p$ для узла p:

1. Если p — инициатор:
   1.1 $s_p := \mathbb{t}$;
   1.2 Для всех $q \in Neigh_p$: $\mathrm{send}_q(\textbf{wakeup})$
2. Пока $r_p \neq Neigh_p$:
   2.1 $\mathrm{receive}_q(\underline{\textbf{wakeup}})$ от любого $q \in Neigh_p$
   2.2 $r_p := r_p \cup \{q\}$;
   2.3 Если $\neg s_p$:
      2.3.1 $s_p := \mathbb{t}$;
      2.3.2 Для всех $q \in Neigh_p$: $\mathrm{send}_q(\textbf{wakeup})$

Фаза волны $Wave_p$ для узла p:

1. Пока $|X_p| > 1$:
   1.1 $\mathrm{receive}_q(\underline{\mathbf{pack}}, v)$ для любого $q \in Neigh_p$
   1.2 $v_p := \min(v_p, v)$;
   1.3 $X_p := X_p \setminus \{q\}$;
2. Пусть $X_p = \{q_0\}$
3. $\mathrm{send}_{q_0}(\mathbf{pack}, v_p)$
4. $\mathrm{receive}_{q_0}(\underline{\mathbf{pack}}, v)$
5. $v_p := \min(v_p, v)$;
6. Для всех $q \in Neigh_p \setminus \{q_0\}$: $\mathrm{send}_q(\mathbf{pack}, v_p)$
7. Если $v_p = v$, то leader, иначе lost

Утверждение (успешность выборов*). В любой достижимой заключительной конфигурации любого вычисления $\mathcal{A}$ узел с наименьшим идентификатором избран, а остальные проиграли

Доказательство.

Применяя те же рассуждения, что и при доказательстве корректности древесного волнового алгоритма, несложно показать, что каждый узел:

- В фазе пробуждения отправляет каждому соседу и принимает от него ровно одно пробуждающее сообщение, после чего переходит к фазе волны
- В фазе волны
  - принимает пакеты от всех соседей, причём так, что последний приём является следствием действий всех узлов сети, и
  - после приёма всех пакетов и соответствующего обновления (5) переменной v_p имеет в ней значение наименьшего идентификатора всех узлов сети

Следовательно, рано или поздно выполняется (7), и справедливость равенства $v_p = v$ при выполнении этого равносильна тому, что узел должен быть избран (и согласно (7) он избирается, а иначе проигрывает) ▼

Утверждение (однородность*). **Алгоритм $\mathcal{A}$ однороден***

Это утверждение очевидно следует из описани $\mathcal{A}$

Утверждение (завершаемость). **Все вычисления с.п. р.с. алгоритма $\mathcal{A}$ конечны**

Это утверждение доказывается приблизительно так же, как и аналогичные до этого

Теорема. **Алгоритм $\mathcal{A}$ является алгоритмом избрания лидера***

Теорема. **Алгоритм $\mathcal{A}$ имеет коммуникационную сложностью $\Theta(n)$ относительно числа узлов n дерева топологии**

Доказательство.

По каждому каналу передаются

- два пробуждающих сообщения в фазе пробуждения и
- два пакета в фазе волны

Следовательно, алгоритм имеет коммуникационную сложность $4m$ для числа каналов m, то есть $4n - 4$ ▼

Диаметр графа — это наибольшее расстояние между его вершинами

Теорема. Алгоритм $\mathcal{A}$ имеет сложность по времени не более $(3\mathfrak{d} + 1)$ относительно диаметра $\mathfrak{d}$ дерева топологии

Доказательство.

В фазе пробуждения узел, находящийся на расстоянии $\mathfrak{d}$ от инициатора, принимает пробуждающие сообщения спустя не более $\mathfrak{d}$ единиц времени, после чего, быть может, рассылает пробуждающие сообщения, доставляющиеся ещё единицу времени — значит, все узлы переходят в фазу волны за время $\mathfrak{d} + 1$

В фазе волны узел, находящийся на расстоянии $\mathfrak{d}$ от листа,

- получает пакет по пути от этого листа спустя не более чем $\mathfrak{d}$ единиц времени и
- передаёт последний пакет всем узлам на пути до этого листа спустя не более $\mathfrak{d}$ единиц времени после принятия решения

Следовательно, сложность $\mathcal{A}$ по времени не превосходит $(3\mathfrak{d} + 1)$ ▼

Д.з. 1. Доказать, что сложность алгоритма $\mathcal{A}$ по времени не превосходит $2\mathfrak{d}$, где $\mathfrak{d}$ — диаметр дерева топологии