

Распределенные алгоритмы и системы

mk.cs.msu.ru → Лекционные курсы → Распределенные алгоритмы и системы

Блок 22

Применение волновых алгоритмов:

PIF,
SYN,
INF

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

PIF

Задача PIF (propagation of information with feedback) широковещательного распространения информации по сети с подтверждением завершения связи устроена так:

- ▶ Задан узел p , обладающий данными d
- ▶ Требуется
 - ▶ (PI) доставить данные d во все узлы сети и
 - ▶ (F) после этого уведомить узел p о том, что данные доставлены
 - ▶ Уверенность узла p в том, что данные доставлены, обозначим командой *done*

PIF-алгоритм — это распределённый алгоритм решения задачи PIF

PIF

Утверждение. Если $done = decide$, то PIF-алгоритм является волновым алгоритмом

Доказательство.

Свойства завершаемости и принятия решения PIF-алгоритма выполняются по определению PIF-алгоритма

Осталось убедиться в полноте покрытия

Предположим от противного, что это свойство не выполняется

Это означает, что существуют вычисление π , действие принятия решения $\vec{\alpha}_\pi[i]$ и узел q , такие что в ни для одного действия $\vec{\alpha}_\pi[j]$ узла q не верно $j \preceq i$

Тогда по теореме о перестановке действий существует и вычисление, в котором сначала принимается решение и только после этого выполняются все действия в узле q

Значит, существует вычисление алгоритма, в котором уведомление о доставке данных происходит до доставки в q — то есть рассматриваемый алгоритм не является PIF-алгоритмом (противоречие) ▼

PIF

Утверждение. Любой централизованный волновой алгоритм, в котором решение принимает инициатор, может быть преобразован в PIF-алгоритм с той же схемой пересылки сообщений

Доказательство.

Для преобразования произвольного централизованного волнового алгоритма в PIF-алгоритм достаточно сделать следующее:

- ▶ Инициатору к каждому отправляемому сообщению добавить рассылаемые данные d
- ▶ Последователю
 - ▶ При приёме первого сообщения сохранить полученные данные d
 - ▶ Добавить к каждому отправляемому сообщению данные d

Тогда данные d будут доставлены в каждый узел-последователь при выполнении первого действия этого узла

Принятие решения волновым алгоритмом означает, что в каждом узле-последователе выполнено хотя бы одно действие

Следовательно, принятие решения означает, что данные d доставлены во все узлы сети ▼

SYN

SYN (SYNchronization) — это синхронизация узлов сети, согласно которой

- ▶ в каждом узле q множества всех узлов V выделяется действие α_q
- ▶ в каждом узле p заданного подмножества P множества V выделяется действие β_p
- ▶ каждое действие β_p , $p \in P$, должно быть выполнено, и при этом строго позже выполнения каждого действия α_q , $q \in V$

SYN-алгоритм — это распределённый алгоритм решения задачи SYN

Утверждение (Задача 32). Каждый волновой алгоритм можно перестроить в SYN-алгоритм, использующий ту же схему обмена сообщениями

Утверждение (Задача 2). Каждый SYN-алгоритм можно перестроить в волновой алгоритм, использующий ту же схему обмена сообщениями

INF

Точной нижней гранью элементов a и b ЧУМ (X, \leq) называется элемент c этого множества, для которого верно следующее:

- ▶ $c \leq a$ и $c \leq b$ (то есть это нижняя грань)
- ▶ Для любого элемента d , такого что $d \leq a$ и $d \leq b$, верно $d \leq c$ (то есть это нижняя грань, наиболее близкая к исходным элементам)

Нижняя полурешётка — это ЧУМ, для любых двух элементов которого существует точная нижняя грань

Точную нижнюю грань элементов a, b нижней полурешётки обозначим записью $a \downarrow b$

Известно, что точная нижняя грань в нижней полурешётке единственна, и операция её вычисления коммутативна ($a \downarrow b = b \downarrow a$) и ассоциативна ($a \downarrow (b \downarrow c) = (a \downarrow b) \downarrow c = a \downarrow b \downarrow c$)

$\downarrow X$ — так для конечного множества $X = \{a_1, \dots, a_n\}$ обозначим точную нижнюю грань элементов этого множества: $\downarrow X = a_1 \downarrow a_2 \downarrow \dots \downarrow a_n$

INF

Примеры точных нижних граней: \min для множеств чисел; $\&$ для булевых значений; \cap для множеств, если \leq — это \subseteq

INF — это задача вычисления точной нижней грани, устроенная так:

- ▶ Каждый узел q (множества V) обладает элементом a_q нижней полурешётки (X, \leq)
- ▶ В выделенном узле p содержится переменная out
- ▶ Требуется
 - ▶ записать в out значение $\downarrow \{a_q \mid q \in V\}$, и после этого
 - ▶ выполнить команду *done* завершения вычисления в узле p

INF-алгоритм — это распределённый алгоритм решения задачи INF

Утверждение (Задача 3). Каждый волновой алгоритм можно перестроить в INF-алгоритм, использующий ту же схему обмена сообщениями

Утверждение (Задача 4). Каждый INF-алгоритм можно перестроить в волновой алгоритм, использующий ту же схему обмена сообщениями