

Распределенные алгоритмы и системы

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

Блок 28

Алгоритмы избрания лидера:
основные определения и допущения,
волновое избрание лидера

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

Задача избрания лидера в распределённой системе состоит в том, чтобы перевести систему

- ▶ из конфигурации, в которой все узлы находятся в одинаковых состояниях
- ▶ в заключительную конфигурацию, в которой ровно один узел обозначен как **лидер**

Необходимость в решении этой задачи возникает в тех случаях, когда требуется запустить централизованный распределённый алгоритм (алгоритм с ровно одним инициатором) и для этого выбрать инициатора

И во всех других случаях, когда требуется среди множества неразличимых узлов особо выделить один: место сбора данных, место принятия решения, центр управления, ...

Алгоритм избрания лидера — это распределённый алгоритм, обладающий следующими свойствами

1. **Однородность**: системы переходов всех узлов одинаковы
2. **Завершаемость**: все вычисления алгоритма конечны
3. **Успешность выборов**: в любой заключительной конфигурации ровно один узел считает себя лидером, а остальные считают себя **проигравшими** выборы
Будем полагать, что узел
 - ▶ **избран** (считает себя лидером), если он хотя бы раз выполнил команду *leader*, и
 - ▶ **проиграл** (считает себя проигравшим), если он хотя бы раз выполнил команду *lost*

Иногда требование явного выполнения команды *lost* проигравшим узлом опускается (лидер может оповестить все узлы о своём выигрыше, например, при помощи волнового алгоритма), но по умолчанию всё равно будем придерживаться этого требования

Алгоритм избрания лидера с такими ограничениями, как правило, невозможен

Чтобы в этом убедиться, достаточно рассмотреть систему из двух узлов с соединяющим их двусторонним (неориентированным) надёжным каналом

Если канал пересылает сообщения за одинаковое время в обе стороны, то согласно **однородности** существует вычисление системы, в котором узлами выполняются одинаковые последовательности действий

В заключительной конфигурации этого вычисления узлы находятся в одинаковых состояниях, поэтому если один из них считает себя лидером, то и второй тоже (что противоречит **успешности выборов**)

Поэтому в алгоритмах избрания лидера принято ослаблять требование однородности, внося в узлы (достаточно малые и естественные) различия в начальных знаниях

В ослабленном требовании однородности (**однородность***) будем допускать следующие различия узлов:

1. Каждому узлу сопоставлен **идентификатор**, являющийся элементом линейно упорядоченного множества
Разным узлам сопоставлены разные идентификаторы
Каждый узел знает свой идентификатор
Идентификаторы разрешено только сравнивать ($=$, $<$)
2. Некоторые узлы объявлены **инициаторами**
Множество инициаторов произвольно, непусто и неизвестно узлам сети
Алгоритм избрания лидера должен корректно выполняться для каждого множества инициаторов
Первым действием не-инициатора (**последователя**) обязан быть приём сообщения (*как и в волновом алгоритме*)
Таким образом, в узлах сети содержатся системы переходов двух видов: одного — в инициаторах, другого — в последователях

В рассматриваемых алгоритмах избрания лидера также будут использоваться, кроме того, ещё и такие дополнительные **допущения и соглашения**:

- ▶ В системе используется асинхронный обмен сообщениями
- ▶ В алгоритме не используется глобальное время, и сообщения могут доставляться сколь угодно долго
- ▶ Каждое сообщение имеет размер $O(W)$, где W — размер записи идентификатора
 - ▶ То есть наибольший размер сообщения не зависит от размера сети
 - ▶ Это соглашение будет использоваться в основном для простоты сравнения сложности разных алгоритмов
- ▶ Лидером *по умолчанию* будет избираться узел с наименьшим значением идентификатора
Можно выбрать лидера и по-другому, и в частности, избранием лидера **среди инициаторов** будем называть избрание узла со значением идентификатора, наименьшим среди всех инициаторов

Для решения задачи избрания лидера можно использовать децентрализованное решение задачи INF:

1. Вычислить в каждом узле точную нижнюю грань *inf* идентификаторов всех узлов
2. Узел избирается, если *inf* совпадает с идентификатором узла, а иначе проигрывает

Задача 1. Покажите (*детальнее*), как можно решить задачу избрания лидера при помощи децентрализованного алгоритма решения задачи INF с произвольным непустым множеством инициаторов, и докажите, что если $decide = leader$, то получившийся алгоритм избрания лидера является волновым

Задача 2. Преобразуйте фазовый алгоритм в алгоритм избрания лидера и посчитайте его коммуникационную и битовую сложности