

Распределённые алгоритмы

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

Блок 23

Сложность распределённого алгоритма
по времени

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

Для вычисления «обычного» (нераспределённого) алгоритма несложно вводится понятие **сложности по времени** работы: например, это может быть

- ▶ длина вычисления (количество шагов) или
- ▶ сумма весов действий, выполняющихся в вычислении
 - ▶ Например, действия могут быть поделены на учитываемые в сумме (веса 1) и не учитываемые (веса 0)

Вычисление р.а. записывается последовательно, но оценивать сложность этого вычисления как длину или как сумму весов в корне неверно:

- ▶ хотя в вычислении действия и записаны последовательно,
- ▶ но «в реальности» некоторые из них могут выполняться параллельно независимо,
- ▶ и если таких действий много, то это может сильно влиять на сложность вычисления

Попробуем ввести понятие сложности вычисления по времени для р.а. так, чтобы в нём

- ▶ был полноценно учтён параллелизм выполнения действий и
- ▶ с учётом этого параллелизма и обязательной последовательности (причинно-следственных связей) подсчитывались время, затрачиваемое на передачу сообщений, если доставка каждого сообщения занимает единицу времени
- ▶ Т.е. собственно действия узлов в этой сложности считаются «бесплатными» — выполняющимися за время, пренебрежимо малое по сравнению со временем доставки сообщений

Сложностью по времени вычисления с.п. р.с. будем называть количество k действий отправки в длиннейшей последовательности действий $\alpha_1^!, \alpha_1^?, \alpha_2^!, \alpha_2^?, \dots, \alpha_k^!, \alpha_k^?$, в которой:

- ▶ Все $\alpha_i^!$ — действия отправки и $\alpha_i^?$ — действия приёма, $i \in \{1, \dots, k\}$
- ▶ Каждое $\alpha_i^!$ взаимосвязано $\alpha_i^?$, $i \in \{1, \dots, k\}$
- ▶ Каждое $\alpha_i^?$ является причиной действия $\alpha_{i+1}^!$, $i \in \{1, \dots, k-1\}$

Пояснение:

- ▶ действие-**следствие** обязательно выполняется (хронологически) после действия-**причины**
- ▶ **несравнимые** действия могут выполняться независимо (параллельно)
- ▶ то есть в этой сложности подсчитывается количество обязательно следующих друг за другом доставок сообщений