

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 подсчитывались действия, относящиеся к передаче сообщений
 - ▶ Т.е. внутренние действия узлов в этой сложности будут считаться «бесплатными» — выполняющимися за время, пренебрежимо малое по сравнению со временем передачи сообщений

Сложностью по времени вычисления с.п. р.с. будем называть длину k длиннейшей последовательности отправок сообщений $\alpha_1^!, \alpha_2^!, \dots, \alpha_k^!$, в которой каждая следующая является **следствием** предыдущей:

$$\alpha_1^! \prec \alpha_2^! \prec \dots \prec \alpha_k^!$$

Пояснение:

- ▶ действие-**следствие** обязательно выполняется (хронологически) строго после действия-**причины**
- ▶ **несравнимые** действия могут выполняться независимо (параллельно)
- ▶ отправка сообщения считается в сложности независимо от наличия взаимосвязанного приёма