Распределенные алгоритмы и системы

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

Блок 3

Модель распределённой системы:
система переходов системы,
система переходов узла,
распределённый алгоритм,
асинхронный и синхронный обмен сообщениями

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

ВМК МГУ, 2022/2023, весенний семестр

Модель распределённой системы

Математическая модель распределённой обработки информации, используемая при изучении распределённых алгоритмов, зависит от выбора исследуемой задачи и сопутствующих целей и средств

Выбор конкретной модели определяется тем, обладает ли она основными необходимыми свойствами:

- Точность, чтобы оценивать сложность алгоритмов и получать результаты о **не**существовании алгоритмов решения рассматриваемых задач
- Общность, чтобы можно было применять её при исследовании достаточно широкого спектра распределённых систем с общими характерными чертами устройства
- Лаконичность, чтобы можно было разумно и обозримо использовать её при проведении доказательств

Модель распределённой системы

Вычисление распределённой системы обычно можно представить как упорядоченную дискретную совокупность действий (событий), выполняющихся (реализующихся; происходящих) в её узлах и отвечающих небольшим изменениям конфигурации системы (её глобального состояния)

Конфигурация системы представляет собой совкупность локальных состояний её узлов и состояния её коммуникационной подсистемы

Способ изменения конфигураций действиями задаётся в виде семейства переходов, аналогичных переходам в конечном автомате и отвечающих влиянию выполнения действий на локальное изменения частей конфигурации, доступных узлам для изменения

Система переходов

Системой переходов (с.п.) будем называть тройку $(\mathcal{C}, \mathcal{I}, \to)$, где:

- $\mathcal{C}$ — непустое множество конфигураций
- $\mathcal{I} \subseteq \mathcal{C}$ — подмножество начальных конфигураций
- $\to \subseteq \mathcal{C} \times \mathcal{C}$ — множество переходов

Систему переходов можно понимать как размеченный ориентированный граф:

- возможно, бесконечный
- с петлями и без кратных дуг
- $\mathcal{C}$ — множество вершин
- вершины множества $\mathcal{I}$ помечены как начальные
- $\to$ — множество дуг

В связи с этим будем применять к с.п. известную графовую терминологию

Система переходов

Частичном вычислением с.п. $S = (\mathcal{C}, \mathcal{I}, \to)$ будем называть путь в S, исходящий из начальной конфигурации

Конфигурацию с.п. назовём заключительной (или, по-другому, тупиковой), если из неё не исходит ни одной дуги

Путь в с.п. будем называть максимальным, если он бесконечен или оканчивается в заключительной конфигурации

(Полным) вычислением с.п. будем называть её максимальное частичное вычисление

Для конфигураций γ_1 и γ_2 с.п. $(\mathcal{C}, \mathcal{I}, \to)$ записью $\gamma_1 \to^* \gamma_2$ будем обозначать факт достижимости γ_2 из γ_1 (то есть $\to^*$ — рефлексивно-транзитивное замыкание отношения $\to$)

Конфигурацию будем называть достижимой в с.п., если она достижима в этой с.п. хотя бы из одной начальной конфигурации

Мультимножество

$\mathbb{N}_0$ — так будем обозначать множество всех целых неотрицательных чисел: $\mathbb{N}_0 = \{0, 1, 2, \dots\}$

Для множества X мультимножеством с основой X называется отображение $M : X \to \mathbb{N}_0$

Содержательно, мультимножество — это неупорядоченная совокупность элементов, в которой (в отличие от «просто» множества) каждый элемент может встречаться произвольное число раз

Значение $M(x)$ для мультимножества M и элемента x называется кратностью (количеством вхождений) x в M

Если $M(x) = 0$, то элемент x не входит в мультимножество M

$\mathbb{M}(X)$ — так обозначим семейство всех мультимножеств с основой X

Конечное мультимножество можно задать списком элементов, как и конечное множество: $\{x_1, x_2, \dots, x_n\}$, где элементы x_i могут повторяться, и каждый элемент записан столько раз, какова его кратность

Мультимножество

Основные операции и отношения над мультимножествами M, N с одной основой X:

- $x \in M \quad \Leftrightarrow \quad M(x) > 0$
- $M \cup N$ — объединение мультимножеств:
  $(M \cup N)(x) = \max(M(x), N(x))$
- $M + N$ — сумма мультимножеств: $(M + N)(x) = M(x) + N(x)$
- $M \cap N$ — пересечение мультимножеств: $(M \cap N) = \min(M(x), N(x))$
- $M - N$ — разность мультимножеств: $(M - N)(x) = M(x) \dot{-} N(x)$, где $\dot{-}$ — операция усечённой разности:
  $$i \dot{-} j = \begin{cases} i - j, & \text{если } i \geq j; \\ 0 & \text{иначе} \end{cases}$$

Пример: если $M = \{a, a, a, b, b, b, b, b, c\}$ и $N = \{a, a, c, c, c, c\}$, то
$$((M \cap N) + M) - (M \cup N) = \{a, a\}$$

Система переходов узла

$\mathcal{M}$ — так будем обозначать множество сообщений, используемых в системе

Будем использовать следующие записи для обозначения обмена сообщениями:

- $m!$, где $m \in \mathcal{M}$ — отправка сообщения m (в коммуникационную подсистему)
- $m?$, где $m \in \mathcal{M}$ — приём сообщения m
- λ — отсутствие взаимодействия с коммуникационной подсистемой

$\mathcal{M}?!$ — так обозначим множество $\{m! \mid m \in \mathcal{M}\} \cup \{m? \mid m \in \mathcal{M}\} \cup \{\lambda\}$

Система переходов узла

Системой переходов узла (с.п.у.) над множеством сообщений $\mathcal{M}$ назовём систему $(Z, I, \mapsto)$, где:

- Z — множество состояний
- $I \subseteq Z$ — подмножество начальных состояний
- $\mapsto \subseteq Z \times \mathcal{M}?! \times Z$ — множество действий

Действие (s, σ, s') будем понимать как дугу графа, помеченную символом σ: $s \overset{\sigma}{\mapsto} s'$

В изображении дуги $s \overset{\lambda}{\mapsto} s'$ будем иногда опускать «λ»: $(s \overset{\lambda}{\mapsto} s') = (s \mapsto s')$

Действие, помеченное отправкой сообщения, приёмом сообщения и символом λ, будем называть соответственно действием отправки, действием приёма и внутренним действием

С.п.у. — это вариант с.п. с доразмеченными дугами и другими названиями компонентов для лучшего различения этих видов с.п.:

- В системе (глобально) — «конфигурация» и «переход»
- В узле (локально) — «состояние» и «действие»

Система переходов узла

Пример псевдокода для узла и соответствующей с.п.у.

$$
\begin{array}{l}
\textbf{var}\ m : bool = \mathbb{f};\\
\textbf{do}\ \{\\
\quad 1\colon\ m := \neg m;\\
\quad 2\colon\ \mathrm{send}(m);\\
\}\ \textbf{until}\ \mathbb{f}
\end{array}
$$

Значение команд и выражений:

- $\textbf{var}\ x : T = v;$: объявление переменной x типа T в узле со значением v в начале выполнения
- $bool$: булев тип с доменом $\{\mathbb{t}, \mathbb{f}\}$ (соответственно истина и ложь)
- $\textbf{do}\ \{\pi\}\ \textbf{until}\ C$: выполнять в цикле π, пока не станет истинным условие C, после чего перейти к выполнению следующих команд
- $x{:=}e$: присвоить значение выражения e в переменную x
- $\neg$: операция отрицания
- $\mathrm{send}(e)$: отправить сообщение, являющееся значением выражения e

Система переходов узла

Пример псевдокода для узла и соответствующей с.п.у.

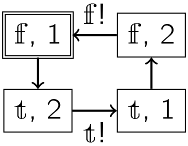
Множество состояний с.п.у. — это множество всевозможных наборов значений переменных узла вместе с номером следующей исполняемой команды (*значением счётчика команд*)

Команды отправки сообщений (send) отвечают действиям отправки с.п.у., команды приёма ($\mathrm{receive}$) — действиям приёма с.п.у., остальные команды — внутренним действиям с.п.у.

Состояния с.п.у. будем изображать окружностями и прямоугольниками, а начальные состояния — двойными окружностями и двойными прямоугольниками

С.п.у. справа отвечает псевдокоду слева

var $m : bool = \mathbb{f};$
do $\{$
  1: $m := \neg m;$
  2: $\mathrm{send}(m);$
$\}$ **until** $\mathbb{f}$



Распределённый алгоритм

Распределённым алгоритмом будем называть конечный набор с.п.у. (по одной с.п.у. для каждого узла системы)

С.п. $(\mathcal{C}, \mathcal{I}, \to)$, распределённого алгоритма $(p_1, \ldots, p_n)$ над множеством сообщений $\mathcal{M}$, где $p_i = (Z_i, I_i, \mapsto_i)$, устроена следующим образом:

- $\mathcal{C} = Z_1 \times \cdots \times Z_n \times \mathbb{M}(\mathcal{M})$
- $\mathcal{I} = I_1 \times \cdots \times I_n \times \{\emptyset\}$
- Устройство множества $\to$ зависит от устройства коммуникационной подсистемы, то есть от того, как именно узлы способны обмениваться сообщениями

Распределённый алгоритм

Два основных способа обмена сообщениями:

- Асинхронный: отправка и приём сообщения происходят независимо, то есть на разных переходах модели
- Синхронный: отправка и приём сообщения происходят одновременно физически или логически, то есть за один переход в модели

Синхронный обмен сообщениями можно представить себе как асинхронный с дополнительными механизмами синхронизации, накладывающими следующее ограничение на поведение системы: после отправки сообщения следующим действием обязательно должен быть приём этого сообщения

Распределённый алгоритм с асинхронным обменом сообщениями

Действие (s, σ, s') k-го узла будем называть допустимым в конфигурации $\gamma = (s_1, \ldots, s_n, M)$, если $s = s_k$ и верно одно из двух:

- $\sigma = \lambda$ или $\sigma = m!$
- $\sigma = m?$ и $m \in M$

Результатом выполнения $\alpha^{(k)}(\gamma)$ допустимого действия $\alpha = (s, \sigma, s')$ k-го узла в конфигурации $\gamma = (s_1, \ldots, s_n, M)$ является конфигурация $(s_1, \ldots, s_{k-1}, s', s_{k+1}, \ldots, s_n, M')$, где

- $M' = M$, если $\sigma = \lambda$
- $M' = M + \{m\}$, если $\sigma = m!$
- $M' = M - \{m\}$, если $\sigma = m?$

Распределённый алгоритм с асинхронным обменом сообщениями

Отношение переходов с.п. алгоритма $(p_1, \ldots, p_n)$ над сообщениями $\mathcal{M}$ с асинхронным обменом — это объединение $\to_1 \cup \cdots \cup \to_n$, где $\to_k$, $1 \le k \le n$ — отношение, описывающее изменение конфигурации системы в результате выполнения действий k-й с.п.у.

Пусть $p_k = (Z, I, \mapsto)$

Тогда отношение $\to_k$ состоит из всех пар вида

$$\gamma \to_k \alpha^{(k)}(\gamma),$$

где γ — конфигурация и $\alpha \in \mapsto$ — допустимое в ней действие

Распределённый алгоритм с асинхронным обменом сообщениями

Пример

Узел A:

$\textbf{var}\ m : bool = \mathbb{f};$
$\textbf{do}\ \{$
  1: $m := \neg m;$
  2: $\mathrm{send}(m);$
$\}\ \textbf{until}\ \mathbb{f}$

Узел B:

$\textbf{var}\ m : bool = *;$
$\textbf{do}\ \{$
  1: $\mathrm{receive}(m);$
$\}\ \textbf{until}\ \mathbb{f}$

Значение команд и выражений:

- $*$: *неопределённое* начальное значение, то есть допускается любое значение в качестве начального
- $\mathrm{receive}(x)$: дождаться, пока в коммуникационной подсистеме сообщение, и принять его, записав в переменную x

Распределённый алгоритм с асинхронным обменом сообщениями

Пример

Узел A:

$\textbf{var}\ m : bool = \mathbb{f};$
$\textbf{do}\ \{$
  $1\colon m := \neg m;$
  $2\colon \mathrm{send}(m);$
$\}\ \textbf{until}\ \mathbb{f}$

Узел B:

$\textbf{var}\ m : bool = *;$
$\textbf{do}\ \{$
  $1\colon \mathrm{receive}(m);$
$\}\ \textbf{until}\ \mathbb{f}$

Одно из вычислений с.п. этого распределённого алгоритма (A, B):

$(\langle \mathbb{f}, 1\rangle, \langle \mathbb{f}, 1\rangle, \emptyset) \to_1 (\langle \mathbb{t}, 2\rangle, \langle \mathbb{f}, 1\rangle, \emptyset) \to_1 (\langle \mathbb{t}, 1\rangle, \langle \mathbb{f}, 1\rangle, \{\mathbb{t}\}) \to_2$
$(\langle \mathbb{t}, 1\rangle, \langle \mathbb{t}, 1\rangle, \emptyset) \to_1 (\langle \mathbb{f}, 2\rangle, \langle \mathbb{t}, 1\rangle, \emptyset) \to_1 (\langle \mathbb{f}, 1\rangle, \langle \mathbb{t}, 1\rangle, \{\mathbb{f}\}) \to_2$
$(\langle \mathbb{f}, 1\rangle, \langle \mathbb{f}, 1\rangle, \emptyset) \to_1 \ldots$

Есть и другое вычисление (и ещё бесконечно много других):

$(\langle \mathbb{f}, 1\rangle, \langle \mathbb{f}, 1\rangle, \emptyset) \to_1 (\langle \mathbb{t}, 2\rangle, \langle \mathbb{f}, 1\rangle, \emptyset) \to_1 (\langle \mathbb{t}, 1\rangle, \langle \mathbb{f}, 1\rangle, \{\mathbb{t}\}) \to_1$
$(\langle \mathbb{f}, 2\rangle, \langle \mathbb{f}, 1\rangle, \{\mathbb{t}\}) \to_1 (\langle \mathbb{f}, 1\rangle, \langle \mathbb{f}, 1\rangle, \{\mathbb{t}, \mathbb{f}\}) \to_1 (\langle \mathbb{t}, 2\rangle, \langle \mathbb{f}, 1\rangle, \{\mathbb{t}, \mathbb{f}\}) \to_1$
$(\langle \mathbb{t}, 2\rangle, \langle \mathbb{f}, 1\rangle, \{\mathbb{t}, \mathbb{t}, \mathbb{f}\}) \to_1 \ldots$

Распределённый алгоритм с синхронным обменом сообщениями

Отношение переходов с.п. алгоритма $(p_1, \ldots, p_n)$ над сообщениями $\mathcal{M}$ с синхронным обменом — это объединение $\bigcup\limits_{1 \le i \le n} \rightarrow_i \cup \bigcup\limits_{\substack{1 \le i \le n \\ 1 \le j \le n \\ i \ne j}} \rightarrow_{i,j}$, где

- $\rightarrow_k$, $1 \le k \le n$, — отношение, описывающее изменение конфигурации системы в результате выполнения **внутренних** действий k-й с.п.у.

- $\rightarrow_{k,m}$, $1 \le k \le n$, $1 \le \ell \le n$, $k \ne \ell$, — отношение, описывающее изменение конфигурации системы в результате одновременного выполнения действий отправки и приёма сообщения разными с.п.у.

Распределённый алгоритм с синхронным обменом сообщениями

Пусть $p_k = (Z_k, I_k, \mapsto_k)$ и $p_\ell = (Z_\ell, I_\ell, \mapsto_\ell)$

Тогда

- отношение $\to_k$ состоит из всех пар вида
$$\gamma \to_k \alpha^{(k)}(\gamma),$$
где γ — конфигурация и $\alpha \in \mapsto$ — допустимое в ней внутреннее действие
- отношение $\to_k$ состоит из всех пар вида
$$\gamma \to_{k,\ell} \alpha_1^{(k)}(\alpha_2^{(\ell)}(\gamma)),$$
где γ — конфигурация $\alpha_1 = (s_k, m!, s'_k) \in \mapsto_k$, $\alpha_2 = (s_\ell, m?, s'_\ell) \in \mapsto_\ell$ и внутренние действия (s_k, s'_k) и (s_ℓ, s'_ℓ) допустимы в γ

Можно легко убедиться в том, что в любой достижимой конфигурации заданной так с.п. не содержится ни одного сообщения, и поэтому будем опускать мультимножество сообщений в записи конфигурации такой с.п.

Распределённый алгоритм с синхронным обменом сообщениями

Пример

Узел A:

$$\begin{array}{l}
\textbf{var } m : bool = \mathbb{f}; \\
\textbf{do } \{ \\
\quad 1\colon m := \neg m; \\
\quad 2\colon \mathrm{send}(m); \\
\}\textbf{ until } \mathbb{f}
\end{array}$$

Узел B:

$$\begin{array}{l}
\textbf{var } m : bool = *; \\
\textbf{do } \{ \\
\quad 1\colon \mathrm{receive}(m); \\
\}\textbf{ until } \mathbb{f}
\end{array}$$

В с.п. этого распределённого алгоритма (A, B) содержится ровно два вычисления

Первое:

$(\langle\mathbb{f}, 1\rangle, \langle\mathbb{f}, 1\rangle) \rightarrow_1 (\langle\mathbb{t}, 2\rangle, \langle\mathbb{f}, 1\rangle) \rightarrow_{1,2} (\langle\mathbb{t}, 1\rangle, \langle\mathbb{t}, 1\rangle) \rightarrow_1 (\langle\mathbb{f}, 2\rangle, \langle\mathbb{t}, 1\rangle) \rightarrow_{1,2} (\langle\mathbb{f}, 1\rangle, \langle\mathbb{f}, 1\rangle) \rightarrow_1 \ldots$

Второе:

$(\langle\mathbb{f}, 1\rangle, \langle\mathbb{t}, 1\rangle) \rightarrow_1 (\langle\mathbb{t}, 2\rangle, \langle\mathbb{t}, 1\rangle) \rightarrow_{1,2} (\langle\mathbb{t}, 1\rangle, \langle\mathbb{t}, 1\rangle) \rightarrow_1 (\langle\mathbb{f}, 2\rangle, \langle\mathbb{t}, 1\rangle) \rightarrow_{1,2} (\langle\mathbb{f}, 1\rangle, \langle\mathbb{f}, 1\rangle) \rightarrow \ldots$