Распределённые алгоритмы

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

Блок 38

Задача сохранения снимка сети

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

ВМК МГУ, 2023/2024, весенний семестр

Понятие снимка

Для диагностики сети и в качестве вспомогательного инструмента при решении других задач полезно уметь получать снимок (snapshot) сети: описание того, в каких состояниях находятся её узлы, или, иными словами, текущую конфигурацию вычисления сети

В связи с этим будем считать термины «снимок» и «конфигурация» синонимами

В системе с централизованной синхронизацией компонентов (программа, цифровая схема, набор программ под контролем ОС, ...) средства получения снимка обычно просты и естественны

Понятие снимка

Но если для извлечения снимка применять распределённые алгоритмы, то всё становится не так просто:

- Снимок как набор данных должен появиться в каком-то узле
- Узел знает только «свои» состояния и выполненные действия и всё то, что было прислано в сообщениях
- В снимок входит и состояние коммуникационной подсистемы, и эта информация ещё менее доступна узлам
- Пока сообщения с информацией о снимке доставляются узлу, эта информация устаревает: другие узлы могут изменять (и почти наверняка изменяют) свои состояния

Примеры использования снимков сети

Но всё же средства извлечения снимков сети весьма полезны

Пример 1: отладка

Один из простых способов отладки программы состоит в том, что

- она выполняется в сочетании с отладчиком,
- в нужный момент её выполнение приостанавливается и
- изучается снимок программы в момент приостановки (значения переменных, стек вызовов и т.п.)

Если хочется аналогично отлаживать р.с., то необходимо научиться извлекать и её снимок

Пример 2: восстановление при сбое

Если компонент р.с. выходит из строя и не хочется из-за этого перезапускать вычисление р.а. «с ноля», то можно

- время от времени сохранять снимки сети и
- после сбоя перезапустить р.а. с последнего снимка

Примеры использования снимков сети

Но всё же средства извлечения снимков сети весьма полезны

Пример 3: анализ монотонных свойств

Напоминание: свойство P конфигураций с.п. монотонно, если для любой конфигурации γ и любого перехода $\gamma \to \delta$ с.п. верно $P(\gamma) \Rightarrow P(\delta)$

Если удастся получить *пусть даже устаревший* снимок γ и убедиться, что верно $P(\gamma)$ для монотонного свойства P, то и для текущей конфигурации δ должно быть верно $P(\delta)$

Примеры «полезных» монотонных свойств:

- «Конфигурация является заключительной»
  - В том числе: «алгоритм ещё не дошёл до конца кода, но все узлы заблокировались (не могут выполнить больше ни одного действия)»
- «Интересующее действие было выполнено»
  - (данные доставлены, решение принято, лидер избран, ...)
- «Сообщение потеряно»
  - (если сообщение потеряно, то оно потеряно навсегда)

Коммуникационная осуществимость снимков

Основная трудность в вычислении снимка сети состоит в том, что вычисляемый снимок должен быть реалистичен: он должен отвечать тому, что было или *хотя бы* могло быть в р.с. в интервале времени от начала вычисления снимка до текущей конфигурации

Для примера можно рассмотреть такое вычисление снимка:

1. Соседним узлам p и q отправляется запрос на снимок их (локальных) состояний
2. Узел p
   - принимает запрос,
   - отправляет текущее состояние как часть снимка и
   - отправляет сообщение m узлу q
3. Узел q
   - принимает сообщение m от p,
   - получает запрос и
   - отправляет текущее состояние (после приёма m) как часть снимка

Такой снимок нереалистичен: согласно снимку, узел q принял сообщение m, а узел p его ещё не отправил

Коммуникационная осуществимость снимков

Если по контексту ясны вычисление E р.с. $\mathcal{S}$ и/или соответствующая последовательность действий $\mathfrak{A} = \mathfrak{Act}(E, \mathcal{S})$ с начальной конфигурацией γ_0, то будем обозначать записью

- $\alpha_p^{(i)}$ i-е действие узла p, начиная с $i = 1$
- $\sigma_p^{(i)}$ состояние узла p после выполнения $\alpha_p^{(i)}$

Если в снимок γ входит состояние $\sigma_p^{(i)}$, то действия $\alpha_p^{(k)}$

- для $k \leq i$ будем называть предваряющими состояние $\sigma_p^{(i)}$ и снимок γ,
- а для $k \geq i + 1$ — следующими за состоянием $\sigma_p^{(i)}$ и снимком γ

То есть $\sigma_p^{(i-1)} \xrightarrow{\alpha_p^{(i)}} \sigma_p^{(i)}$, и появление $\sigma_p^{(i)}$ в снимке означает, что в p выполнены все действия, предваряющие $\sigma_p^{(i)}$, и не выполнено ни одно следующее

$\gamma[p]$ — так будем обозначать состояние узла p в снимке γ

Коммуникационная осуществимость снимков

Для технической простоты будем полагать, что состоянием $\gamma[p]$ однозначно задаются

- последовательность $sent^{\gamma}_{p\to q}$ сообщений, отправленных в каждый канал $p \to q$ предваряющими действиями и
- последовательность $recv^{\gamma}_{q\to p}$ сообщений, принятых из каждого канала $q \to p$ предваряющими действиями

Кроме того, для простоты будем считать, что все отправляемые сообщения попарно различны и пронумерованы согласно порядку отправки в вычислении: $m^{(1)}, m^{(2)}, \ldots$

Тогда состояниями $\gamma[p]$ и $\gamma[q]$ однозначно задаётся последовательность $mes^{\gamma}_{p\to q}$ сообщений, отправленных в канал $p \to q$ и ещё не принятых из него, располагающихся согласно очерёдности отправки в канал: это последовательность $sent^{\gamma}_{p\to q} \setminus recv^{\gamma}_{p\to q}$, получающаяся из $sent^{\gamma}_{p\to q}$ удалением всех элементов последовательности $recv^{\gamma}_{p\to q}$

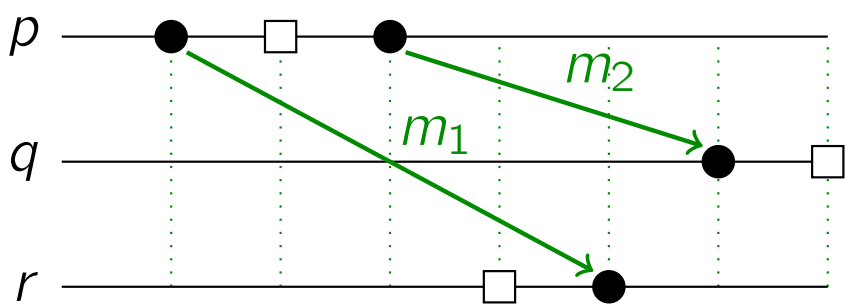
В таких упрощениях для вычисления снимка достаточно вычислить состояния всех узлов в нём, поэтому далее снимком будем называть набор состояний узлов (без состояния коммуникационной подсистемы)

Коммуникационная осуществимость снимков

Снимок γ назовём коммуникационно осуществимым, если для любого канала $p \to q$ верно соотношение $recv^{\gamma}_{p \to q} \subseteq sent^{\gamma}_{p \to q}$: последовательность $recv^{\gamma}_{p \to q}$ является подпоследовательностью последовательности $sent^{\gamma}_{p \to q}$

То есть если все сообщения, которые считаются принятыми, также считаются и отправленными (согласно состояниям узлов в снимке)

Пример коммуникационно неосуществимого снимка γ, изображённого на диаграмме событий:

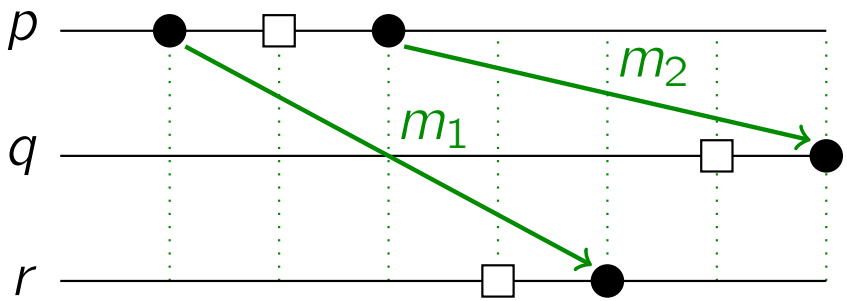


Состояния снимка изображены прямоугольниками, расположенными после предваряющих действий и перед следующими

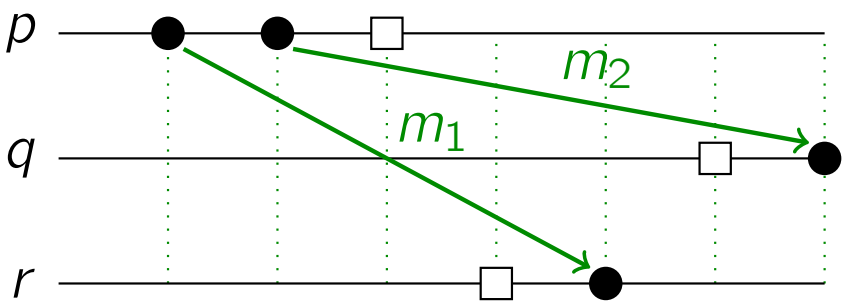
$recv^{\gamma}_{p \to q} = (m_2) \not\subseteq \emptyset = sent^{\gamma}_{p \to q}$

Коммуникационная осуществимость снимков

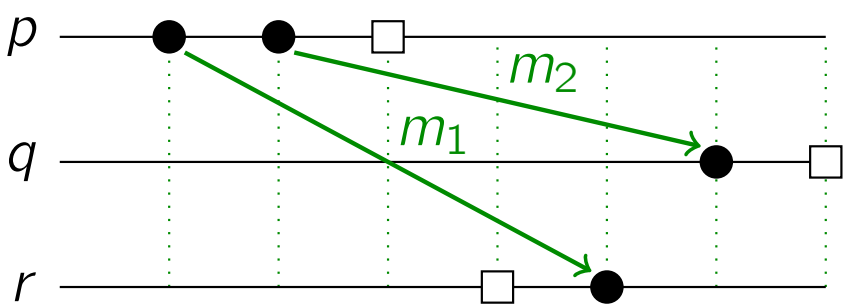
Примеры коммуникационно осуществимых снимков



$$recv^{\gamma}_{p\to q} = \emptyset \subseteq \emptyset = sent^{\gamma}_{p\to q}$$
$$recv^{\gamma}_{p\to r} = \emptyset \subseteq (m_1) = sent^{\gamma}_{p\to q}$$



$$recv^{\gamma}_{p\to q} = \emptyset \subseteq (m_2) = sent^{\gamma}_{p\to q}$$
$$recv^{\gamma}_{p\to r} = \emptyset \subseteq (m_1) = sent^{\gamma}_{p\to q}$$



$$recv^{\gamma}_{p\to q} = (m_2) \subseteq (m_2) = sent^{\gamma}_{p\to q}$$
$$recv^{\gamma}_{p\to r} = \emptyset \subseteq (m_1) = sent^{\gamma}_{p\to q}$$

Сообщение, отправленное действием, предваряющим снимок, будем называть предваряющим сообщением, а отправленное действием, следующим за снимком, будем называть следующим сообщением

Сечения вычислений

Рассмотрим последовательность действий $\mathfrak{A} = \mathfrak{Act}(E, \mathcal{S})$ вычисления E р.с. $\mathcal{S}$

Сечением последовательности $\mathfrak{A}$ и вычисления E будем называть подпоследовательность $\mathcal{L} \subseteq \mathfrak{A}$, в которой любое действие любого узла входит только вместе со всеми выполнившимися ранее действиями этого узла:

$$\forall p \in \mathcal{S} : \forall i, j \in \mathbb{N} : \alpha_p^{(j)} \in \mathcal{L} \,\&\, i < j \Rightarrow \alpha_p^{(i)} \in \mathcal{L}$$

Утверждение (Д.з. 1). Подпоследовательность $\mathcal{L}$ последовательности действий $\mathfrak{A}$ является сечением $\Leftrightarrow$ существует снимок γ, такой что $\mathcal{L}$ состоит в точности из всех действий, предваряющих γ

Сечения вычислений

Рассмотрим последовательность действий $\mathfrak{A} = \mathfrak{Act}(E, \mathcal{S})$ вычисления E р.с. $\mathcal{S}$

$\mathcal{L}(\gamma)$ — так будем обозначать сечение, состоящее из всех действий, предваряющих снимок γ

Сечение $\mathcal{L}_1$ будем называть предшествующим сечению $\mathcal{L}_2$, и $\mathcal{L}_2$ — следующим за $\mathcal{L}_1$, если $\mathcal{L}_1 \subseteq \mathcal{L}_2$

Сечение будем называть согласованным, если каждое действие входит в него только вместе со всеми своими причинами:

$$\forall \alpha\ , \beta \in \mathfrak{A} : \beta \in \mathcal{L}\ \&\ \alpha \prec \beta \Rightarrow \alpha \in \mathcal{L}$$

Сечения вычислений

Д.з. 2. Приведите пример несогласованного сечения $\mathcal{L}$ для вычисления, изображённого ранее на диаграммах событий, и снимка γ, которому соответствует это сечение ($\mathcal{L} = \mathcal{L}(\gamma)$)

Д.з. 3. В блоке 22 рассказывалось про логические часы Лэмпорта θ_L. Обязательно ли (для любого ли $k \in \mathbb{N}$ и для любого ли вычисления) множество всех действий всех узлов вычисления, для которых значения часов Лэмпорта не превосходят k, является (а) сечением, и (б) согласованным сечением?

Значимые снимки

При извлечении снимка *хотелось бы* иметь снимок, представляющий собой конфигурацию вычисления

Но вычисление E имеет смысл рассматривать не само по себе, а только вместе с классом $\mathcal{E}$ всех вычислений, получающихся из него перестановкой действий с сохранением причинно-следственного порядка

Поэтому ослабим требование, предъявляемое к алгоритму сохранения снимка, так, чтобы им вычислялся снимок, являющийся конфигурацией какого-либо вычисления из $\mathcal{E}$, но не обязательно именно вычисления E

(«Даже если γ и не было, то оно с тем же успехом могло бы и быть»)

Снимок γ назовём значимым для вычисления E, если существует вычисление $E' \in \mathcal{E}$, в которое входит γ

Значимые снимки

Теорема (о снимках и сечениях). **Для любого снимка γ следующие три утверждения эквивалентны:**

1. **Снимок γ коммуникационно осуществим**
2. **Сечение $\mathcal{L}(\gamma)$ согласованно**
3. **Снимок γ значим**

Значимые снимки

Доказательство ($1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$). *$\gamma$ осуществим $\Rightarrow$ $\mathcal{L}(\gamma)$ согласованно*

Рассмотрим осуществимый снимок γ, действие $\alpha \in \mathcal{L}(\gamma)$ и причину β этого действия, и покажем, что $\beta \in \mathcal{L}(\gamma)$

По определению $\preceq$, для этого достаточно показать, что $\beta \in \mathcal{L}(\gamma)$, для двух случаев:

1. $\alpha = \alpha_p^{(i)}$ и $\beta = \alpha_p^{(i-1)}$
   Тогда соотношение $\beta \in \mathcal{L}(\gamma)$ верно по определению сечения

2. α — действие приёма сообщения m, а β — взаимосвязанное действие отправки
   Пусть $\alpha \in \mathcal{A}_q$ и $\beta \in \mathcal{A}_p$
   Тогда
   $\alpha \in \mathcal{L}(\gamma) \quad \Rightarrow \quad$ (по определению $\mathcal{L}$)
   $m \in recv_{p \to q}^{\gamma} \quad \Rightarrow \quad$ (т.к. γ коммуникационно осуществим)
   $m \in sent_{p \to q}^{\gamma} \quad \Rightarrow \quad$ (по определению $\mathcal{L}$)
   $\beta \in \mathcal{L}(\gamma)$

Значимые снимки

Доказательство ($1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$). *$\mathcal{L}(\gamma)$ согласованно $\Rightarrow$ γ значим*

Чтобы это показать, достаточно построить вычисление E', отличающееся от исходного E только перестановкой действий, сохраняющей причинно-следственный порядок $\prec$ и содержащее снимок γ

Для этого достаточно организовать E' так, чтобы все действия, предваряющие γ, выполнялись в E' до γ, а следующие за γ — после γ

Рассмотрим следующую перестановку $\mathfrak{A}'$ действий $\mathfrak{A} = \mathfrak{Act}(E, \mathcal{S})$:

- сначала в ней перечислены все предваряющие действия в любом порядке, согласованном с $\prec$ $(\mathfrak{A}'[1], \ldots, \mathfrak{A}'[k])$,
- и затем — все следующие действия в любом порядке, согласованном с $\prec$

Если показать, что перестановка $\mathfrak{A}'$ сохраняет порядок $\prec$, то из теоремы о перестановке действий будет следовать, что это последовательность действий, отвечающая некоторому вычислению (что и требуется)

Д.з. 4: показать, что перестановка $\mathfrak{A}'$ сохраняет $\prec$

Значимые снимки

Доказательство ($1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$). *$\gamma$ значим $\Rightarrow$ γ осуществим*

По определению значимости, γ содержится в некотором вычислении E', отличающемся от E только перестановкой действий, сохраняющей порядок $\prec$

Согласно устройству вычисления с.п., перед приёмом сообщения в вычислении обязательно выполняется его отправка

Значит, для вычисления E' и любого канала $p \to q$ справедливо $recv^{\gamma}_{p \to q} \subseteq sent^{\gamma}_{p \to q}$

Но любая перестановка действий, сохраняющая порядок $\prec$, сохраняет в частности и порядок выполнения действий в узле, а значит, и последовательности сообщений, отправленных узлом в каждый доступный канал и принятых им из каждого доступного канала

Значит, для вычисления E и всех каналов $p \to q$ справедливы те же соотношения $recv^{\gamma}_{p \to q} \subseteq sent^{\gamma}_{p \to q}$ ▼

Алгоритмы сохранения снимка

Алгоритм сохранения снимка — это распределённый алгоритм, надстраивающийся над произвольным заданным алгоритмом и добавляющий возможность после своего запуска сохранить в каждом узле состояние так, чтобы совокупность сохранённых состояний представляла собой значимый снимок

snap — так будем обозначать команду сохранения текущего состояния узла для снимка

Таким образом, обсуждение алгоритма сохранения снимка — это обсуждение двух алгоритмов:

1. собственно алгоритма сохранения снимка
   - (будем называть этот алгоритм, его сообщения, действия и остальные элементы контрольными) и
2. алгоритма, над которым он надстраивается
   - (будем называть этот алгоритм, его сообщения, действия и остальные элементы базовыми)