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, 2025, февраль–май

Вступление

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 — иными словами, текущую конфигурацию вычисления сети

В связи с этим пока 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

Вступление

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

Вступление

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

Вступление

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)

Понятие снимка

Для технической простоты далее будем полагать следующее:

- Все сообщения, отправляющиеся в вычислении р.с., попарно различны (пронумерованы, как и действия вычисления)
- Каждый узел p для каждого соседа q хранит и обновляет два значения:
  - $sent_{p \to q}$ — последовательность сообщений, отправленных узлу q с начала вычисления до текущей конфигурации, в порядке отправки
  - $recv_{q \to p}$ — последовательность сообщений, принятых от q с начала вычисления до текущей конфигурации, в порядке приёма

Тогда конфигурация γ однозначно задаётся набором состояний узлов, и такой набор будем называть снимком, представляющим γ

Если каждое состояние снимка содержится хотя бы в одной конфигурации из π, то такой снимок будем называть снимком вычисления π

Значения $sent_{p \to q}$ и $recv_{q \to p}$ в снимке σ будем обозначать так: $sent^{\sigma}_{p \to q}$ и $recv^{\sigma}_{q \to p}$

Понятие снимка

Пусть заданы р.с. $\mathcal{S}$, её вычисление π и последовательность действий $\mathfrak{A} = \mathfrak{Act}(\pi, \mathcal{S})$

Будем использовать следующие обозначения:

- $\gamma[p]$ — состояние узла p в конфигурации или снимка γ
- $\mathfrak{A}_p^{(n)}$ — n-е действие узла p в $\mathfrak{A}$, начиная с $n = 1$
- $\pi_p^{(n)}$ — состояние узла p в конфигурации вычисления π после выполнения $\mathfrak{A}_p^{(n)}$ (считаем, что $\pi_p^{(0)} = \pi(0)[p]$)

Действие $\mathfrak{A}_p^{(k)}$ будем называть предшествующим состоянию $\pi_p^{(n)}$, если $k \leq n$, и следующим за этим состоянием, если $k > n$

Для каждого узла p действия, предшествующие состоянию $\gamma[p] = \pi_p^{(n)}$ конфигурации или снимка γ или следующие за ним, будем называть также соответственно предшествующими γ или следующими за γ

Значимый снимок

При извлечении снимка *хотелось бы* сделать так, чтобы он представлял собой конфигурацию имеющегося вычисления

Но вычисление π имеет смысл рассматривать только вместе с классом Π всех эквивалентных ему вычислений: получающихся из него перестановкой действий с сохранением причинно-следственного порядка

Вычисления из Π отличаются от π только хронологией независимых событий, которая зачастую неважна (а важны только причинно-следственные зависимости между действиями), и поэтому *разумно и приемлемо* было бы вычислить снимок, представляющий быть может не конфигурацию вычисления π, но хотя бы уж конфигурацию вычисления из Π

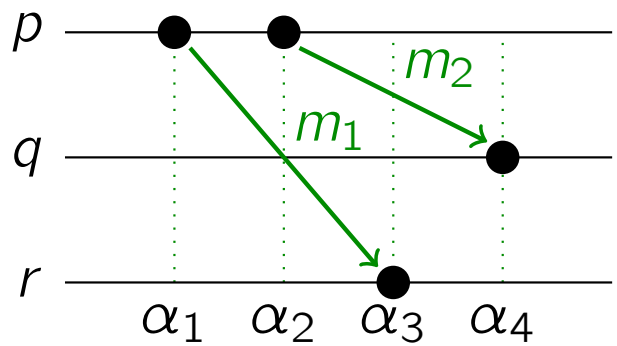
Тогда можно быть уверенным в том, что даже если вычисленного снимка фактически нет в вычислении, но он наверняка возможен для подходящей хронологии независимых событий

Значимый снимок

Пусть заданы р.с. $\mathcal{S}$ и её вычисление π

Снимок σ вычисления π назовём значимым, если σ представляет хотя бы одну конфигурацию хотя бы одного вычисления, эквивалентного π

Д.з. 1. Сколько существует значимых снимков изображённого ниже вычисления?



Коммуникационная осуществимость снимка

Рассмотрим такой сценарий вычисления снимка:

1. Соседним узлам p и q отправляется запрос снимка их (локальных) состояний
2. Узел p
   - принимает запрос снимка,
   - отправляет своё текущее состояние s_p в ответ на запрос и
   - отправляет сообщение m узлу q согласно выполняющемуся р.а.
3. Узел q
   - принимает сообщение m от p согласно выполняющемуся р.а.,
   - принимает запрос снимка и
   - отправляет своё текущее состояние s_q в ответ на запрос

Вычисленный так снимок утверждает, основываясь на s_p и s_q, что q принял m, а p ещё не отправил m

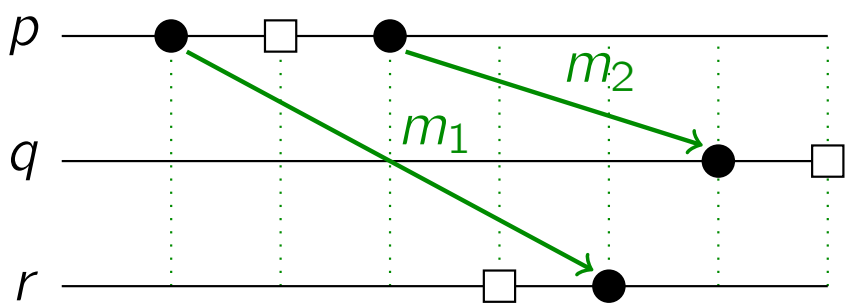
Независимо от значимости, хотелось бы, чтобы из снимка не следовало, что в системе принято сообщение, которое никем не отправлялось

Коммуникационная осуществимость снимка

Снимок σ назовём коммуникационно осуществимым, если для любых смежных узлов p, q верно соотношение $recv^{\sigma}_{p\to q} \subseteq sent^{\sigma}_{p\to q}$: последовательность $recv^{\sigma}_{p\to q}$ является подпоследовательностью последовательности $sent^{\sigma}_{p\to q}$

Пример

Состояния узлов, входящие в снимок, будем изображать на диаграмме событий прямоугольниками, расположенными сразу после всех предшествующих действий соответствующих узлов

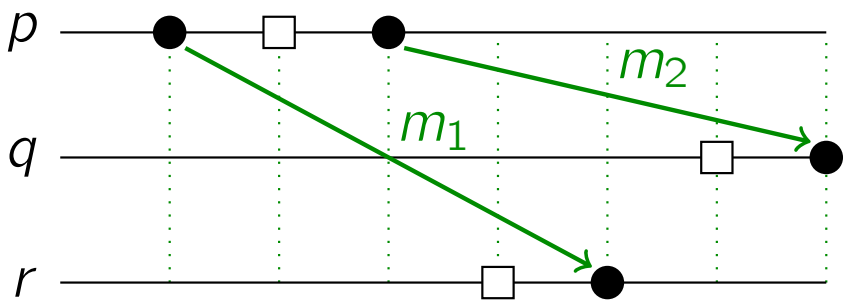


Изображённый снимок σ коммуникационно неосуществим:

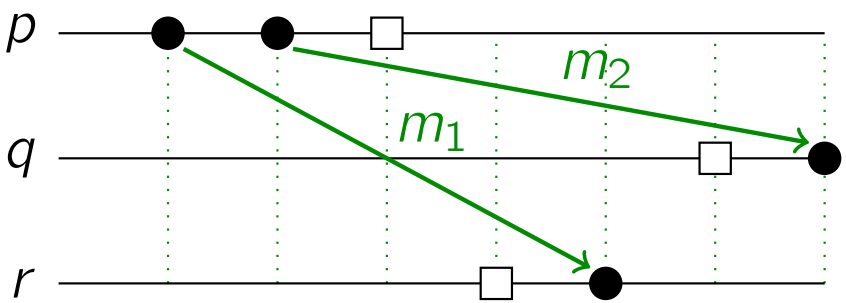
$$recv^{\sigma}_{p\to q} = (m_2) \not\subseteq \emptyset = sent^{\sigma}_{p\to q}$$

Коммуникационная осуществимость снимка

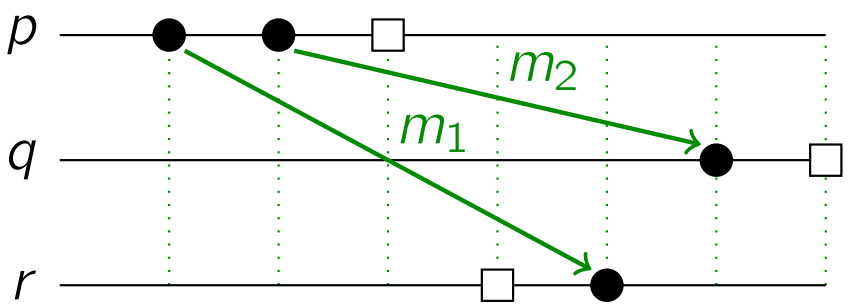
Примеры коммуникационно осуществимых снимков σ



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



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



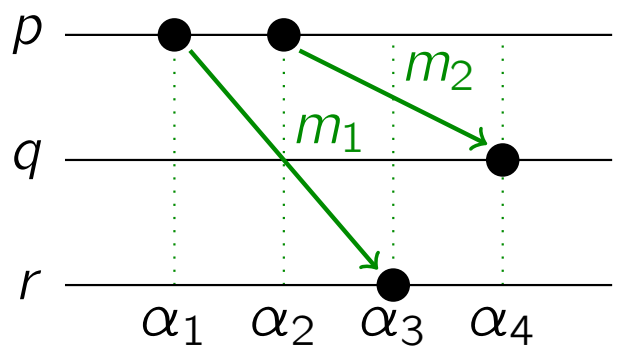
$$recv^{\sigma}_{p\to q} = (m_2) \subseteq (m_2) = sent^{\sigma}_{p\to q}$$
$$recv^{\sigma}_{p\to r} = \emptyset \subseteq (m_1) = sent^{\sigma}_{p\to q}$$

Сечение вычисления

Пусть заданы р.с. $\mathcal{S}$, её вычисление π, последовательность действий $\mathfrak{A} = \mathfrak{Act}(\pi, \mathcal{S})$ и последовательность $\mathcal{L} \subseteq \mathfrak{A}$

Будем называть $\mathcal{L}$ сечением последовательности $\mathfrak{A}$ и вычисления π, если для любого узла p и любого действия $\alpha_p^{(n)}$, входящего в $\mathcal{L}$, все действия $\alpha_p^{(1)}, \ldots, \alpha_p^{(n-1)}$ также входят в $\mathcal{L}$

Пример



Следующие последовательности являются сечениями этого вычисления:

$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ $(\alpha_1, \alpha_2, \alpha_4)$ (α_1, α_2) (α_1, α_3) (α_3) $()$

Не являются сечениями, например, последовательности

$(\alpha_2, \alpha_3, \alpha_4)$ (α_2, α_4) (α_2)

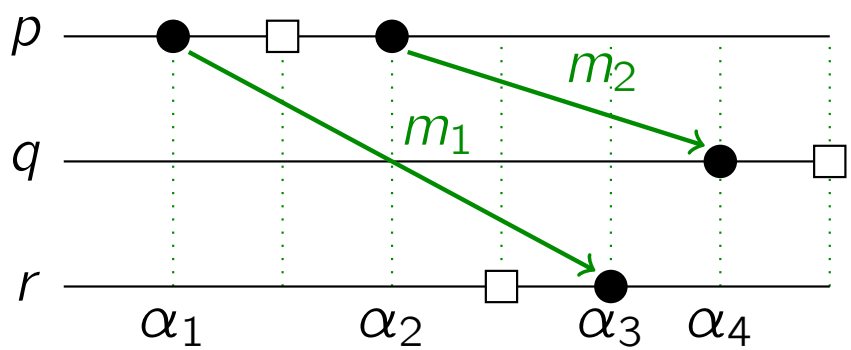
Сечение вычисления

Пусть заданы р.с. $\mathcal{S}$, её вычисление π, последовательность действий $\mathfrak{A} = \mathfrak{Act}(\pi, \mathcal{S})$ и последовательность $\mathcal{L} \subseteq \mathfrak{A}$

Утверждение (Д.з. 2). **$\mathcal{L}$ — сечение вычисления $\pi \Leftrightarrow$ существует снимок σ вычисления π, такой что $\mathcal{L}$ состоит в точности из всех действий, предшествующих σ**

$\mathcal{L}(\sigma)$ — так будем обозначать сечение, состоящее из всех действий, предшествующих снимку σ

Пример



Для изображённого снимка σ верно $\mathcal{L}(\sigma) = (\alpha_1, \alpha_4)$

Сечение вычисления

Пусть заданы р.с. $\mathcal{S}$, её вычисление π, последовательность действий $\mathfrak{A} = \mathfrak{Act}(\pi, \mathcal{S})$ и последовательность $\mathcal{L} \subseteq \mathfrak{A}$

Сечение $\mathcal{L}$ будем называть согласованным, если для любого действия из $\mathcal{L}$ все причины этого действия тоже входят в $\mathcal{L}$

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?

Теорема (о снимках и сечениях)

Пусть заданы р.с. $\mathcal{S}$, её вычисление π и снимок σ этого вычисления. Тогда следующие три утверждения эквивалентны:

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

Теорема (о снимках и сечениях)

Доказательство ($1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$). *$\sigma$ значим $\Rightarrow$ σ осуществим*

По определению значимости, конфигурация γ, представленная снимком σ, содержится в некотором вычислении π', эквивалентном π

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

Значит, для каждого приёма сообщения, предшествующего γ в π', есть и отправка этого сообщения, предшествующая γ в π'

Из этого и того, что снимок σ представляет γ, следует, что для каждой пары смежных узлов верно $recv^{\sigma}_{p \to q} \subseteq sent^{\sigma}_{p \to q}$

Теорема (о снимках и сечениях)

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

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

По определению $\preceq$, достаточно рассмотреть два случая:

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

2. α — действие приёма сообщения m, а β — взаимосвязанное действие отправки
   Пусть α и β — действия узлов q и p соответственно
   Тогда

   $\alpha \in \mathcal{L}(\sigma) \quad \Rightarrow \quad$ (по определению $\mathcal{L}(\sigma)$ и *recv*)
   $m \in recv^{\sigma}_{p \to q} \quad \Rightarrow \quad$ (т.к. σ коммуникационно осуществим)
   $m \in sent^{\sigma}_{p \to q} \quad \Rightarrow \quad$ (по определению $\mathcal{L}(\sigma)$ и *sent*)
   $\beta \in \mathcal{L}(\sigma)$

Теорема (о снимках и сечениях)

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

Рассмотрим такую перестановку $\mathfrak{A}'$ действий $\mathfrak{Act}(\pi, \mathcal{S})$:

- сначала в ней перечислены все действия, предшествующие σ, в любом порядке, согласованном с $\prec$,
- и затем — все действия, следующие за σ, в любом порядке, согласованном с $\prec$

Д.з. 4: показать, что если $\mathcal{L}(\sigma)$ согласованно, то $\mathfrak{A}'$ сохраняет $\prec$

Из сохранения $\prec$ перестановкой $\mathfrak{A}'$ и теоремы о перестановке действий следует, что существует вычисление π', последовательностью действий которого является $\mathfrak{A}'$

Сохранение $\prec$ перестановкой $\mathfrak{A}'$ означает, что π' эквивалентно π

По выбору перестановки $\mathfrak{A}'$, σ представляет конфигурацию $\pi'(n)$, где n — то, сколько действий предшествует снимку σ ▼

Алгоритмы сохранения снимка

Алгоритм сохранения снимка — это распределённый алгоритм, надстраивающийся над произвольным заданным алгоритмом и добавляющий возможность после своего запуска сохранить в каждом узле текущее 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. надстраивающегося алгоритма
   - (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)