Распределенные алгоритмы и системы

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

Блок 34

Задача сохранения снимка сети

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

ВМК МГУ, 2022/2023, весенний семестр

Снимок сети

Иногда (для решения ряда задач с помощью распределённых алгоритмов, и в целом для диагностики сети) возникает желание получить и проанализировать текущую конфигурацию вычисления распределённой системы — или, по-другому, её снимок (snapshot), или моментальное состояние

Иногда снимок сети можно получить извне, если либо существенные его части доступны для наблюдения напрямую, либо имеются подходящие средства извлечения снимка

Но хотелось бы уметь получать снимок сети и изнутри сети, чтобы можно было использовать снимки при проектирвоании распределённых алгоритмов и иметь средства сбора снимков для отправки вовне

Снимок сети

Каждый узел распределённой системы знает своё текущее состояние

Труднее оценить в узле текущие состояния других узлов: такая информация может быть передана в узел только при помощи обмена сообщениями, занимающего некоторое время, и поэтому по достижении узла является заведомо устаревшей

Ещё труднее оценить состояние коммуникационной подсистемы (какие сообщения находятся «в пути», то есть отправлены, но ещё не приняты): для этого требуется получить согласованную информацию и обо всех отправленных сообщениях, и обо всех принятых

Примеры применения снимков

1.

Свойство P конфигураций с.п. $S = (\mathcal{C}, \mathcal{I}, \to)$ называется устойчивым, если переход из конфигурации, обладающей свойством P, возможен только в конфигурацию, также обладающую свойством P:

$$\forall \gamma \ \in \mathcal{C} : \forall \gamma' \in \mathcal{C}, \gamma \to \gamma' : P(\gamma) \Rightarrow P(\gamma')$$

Иными словами, устойчивость — это второй пункт определения инварианта с.п.

Если есть возможность получить конфигурацию сети, то даже если в процессе получения или анализа она устареет, то результат анализа устойчивых свойств снимка останется верен и для всех последующих конфигураций

Примеры применения снимков

Например:

- Если распределённый алгоритм завершил свою выполнение, то оно остаётся завершённым и далее
- Если узлы оказались заблокированы: ожидают приёма сообщения друг от друга перед выполнением других действий (в том числе перед ожидаемой отправкой) — то узлы останутся заблокированными и во всех следующих конфигурациях
- Контроль потери данных: например, если по сети передаётся фишка, как в алгоритме обхода, и эта фишка вдруг исчезает из сети, то она исчезает навсегда
- Сборка мусора: если в сети появляются сообщения, которые согласно алгоритму никто не больше не примет, или в узлах появляются данные, к которым эти узлы не смогут обратиться (*как, например, при потере всех указателей*), то эти сообщения и данные остаются в сети навсегда

Примеры применения снимков

2.

Компоненты распределённой системы могут выходить из строя

Чтобы не перезапускать вычисление алгоритма «с ноля», можно

- время от времени сохранять снимки сети и
- после восстановления узлов, вышедших из строя, установить последний снимок в качестве начальной конфигурации сети

3.

При проектировании распределённых программ требуется их отлаживать

Выполнение «обычной» нераспределённой программы можно приостановить, чтобы в рамках отладки проанализировать снимок, содержащий значения переменных, стек вызовов функций, следующую выполняющуюся команду и т.п.

Точно так же можно использовать для отладки и снимок сети

Осуществимость снимков

Основная трудность в вычислении снимка сети состоит в том, чтобы обеспечить *корректность* (реалистичность) вычисляемого снимка

Для примера можно рассмотреть такую последовательность действий по построению снимка:

1. Смежным узлам p и q отправляется запрос на снимок их (локального) состояния
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

Согласно полученному снимку, узел q принял сообщение m, но узел p ещё его не отправил

Такой конфигурации достичь невозможно, снимок некорректен

Осуществимость снимков

Для узла p распределённого алгоритма $\mathfrak{A}$ и вычисления E этого алгоритма пронумеруем действия узла p в последовательности $Act(E, \mathfrak{A})$ согласно порядку их расположения в последовательности: $\alpha_p^{(1)}, \alpha_p^{(2)}, \ldots$

Состояние узла p сразу после выполнения действия $\alpha_p^{(i)}$ будем обозначать записью $s_p^{(i)}$ (начальное состояние — записью $s_p^{(0)}$)

Для технической простоты будем считать все действия $\alpha_p^{(i)}$ попарно различными, как и все состояния $s_p^{(i)}$

В связи с уникальностью действий будем для $\mathcal{A} = Act(E, \mathfrak{A})$ писать $\mathcal{A}[i] \prec \mathcal{A}[j]$ наряду с $i \prec j$

Осуществимость снимков

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

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

То есть $s_p^{(i-1)} \xrightarrow{\alpha_p^{(i)}} s_p^{(i)}$, и появление $s_p^{(i)}$ в снимке означает, что выполнены все предшествующие действия и не начали выполняться следующие

Осуществимость снимков

Для технической простоты будем полагать, что состоянием $\gamma[p]$ узла p в снимке γ однозначно задаётся

- 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, и что
- все отправляемые сообщения попарно различны и пронумерованы согласно порядку отправки в вычислении: $m^{(1)}, m^{(2)}, \ldots$

Тогда снимком γ для каждого канала $p \to q$ однозначно определяется последовательность $mes^{\gamma}_{p\to q} = sent^{\gamma}_{p\to q} \setminus recv^{\gamma}_{p\to q}$ сообщений, отправленных в канал $p \to q$ и ещё не принятых из него, располагающихся согласно очерёдности отправки в канал: эта последовательность получается из $sent^{\gamma}_{p\to q}$ удалением всех элементов последовательности $recv^{\gamma}_{p\to q}$

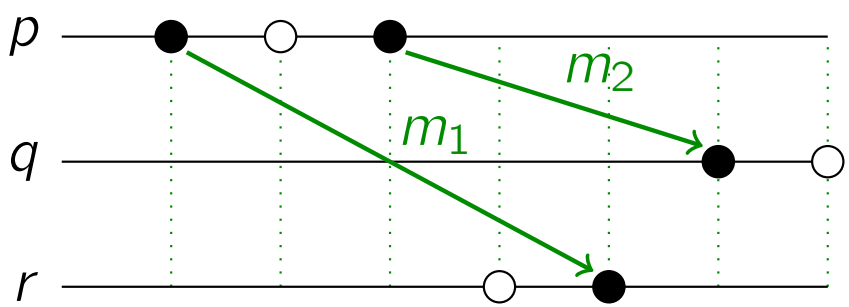
В таких упрощениях вычисление снимка сети означает вычисления набора состояний всех узлов в этом снимке

Осуществимость снимков

Снимок γ называется осуществимым, если для любого канала $p \to q$ верно соотношение $rcvd^{\gamma}_{p\to q} \subseteq sent^{\gamma}_{p\to q}$, то есть если последовательность сообщений, принятых из канала, является подпоследовательностью последовательности сообщений, отправленных в этот канал

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

Пример неосуществимого снимка

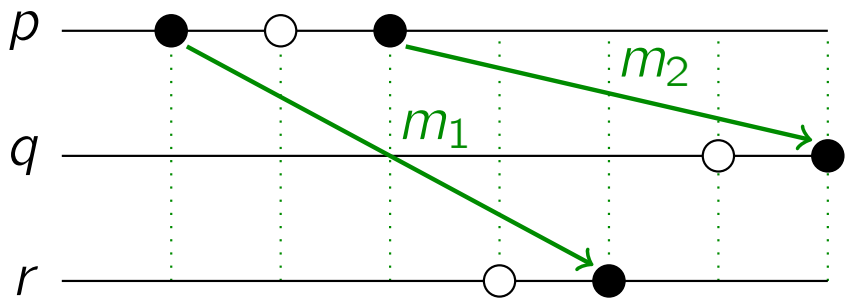


Состояния каждого из узлов p, q, r, входящие в снимок γ, изображены белыми кругами

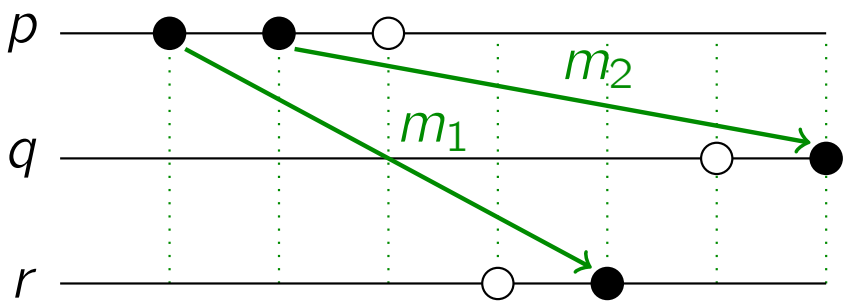
$$recv^{\gamma}_{p\to q} = \{m_2\} \not\subseteq \emptyset = sent^{\gamma}_{p\to q}$$

Осуществимость снимков

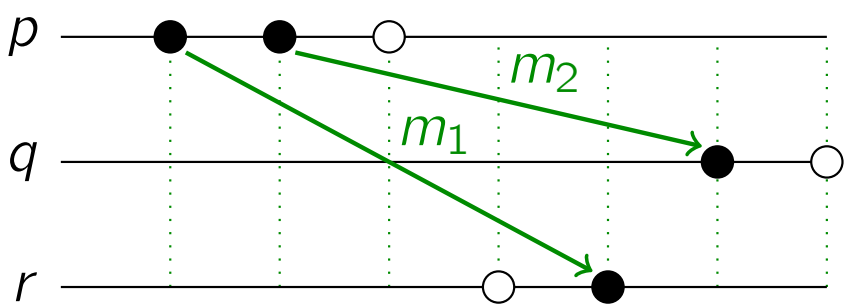
Примеры осуществимых снимков



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



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



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

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

Сечения вычислений

Рассмотрим последовательность действий $\mathcal{A} = Act(E, \mathfrak{A})$, отвечающую вычислению E распределённого алгоритма $\mathfrak{A}$

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

$$\forall p \in \mathfrak{A} : \forall i, j \in \mathbb{N}_0, i < j : \alpha_p^{(j)} \in \mathcal{L} \Rightarrow \alpha_p^{(i)} \in \mathcal{L}$$

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

Сечения вычислений

Рассмотрим последовательность действий $\mathcal{A} = Act(E, \mathfrak{A})$, отвечающую вычислению E распределённого алгоритма $\mathfrak{A}$

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

Сечение $\mathcal{L}_1$ будем называть более ранним по сравнению с $\mathcal{L}_2$, и $\mathcal{L}_2$ — более поздним по сравнению с $\mathcal{L}_1$, если $\mathcal{L}_1 \subseteq \mathcal{L}_2$

Сечение будем называть согласованным, если каждое действие входит в него только вместе со всеми причинно-следственно предшествующими ему действиями:

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

Сечения вычислений

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

Задача 3. В блоке 7 рассказывалось про логические часы, и в частности, часы Лэмпорта ($\theta_L(E, i)$) Для каких $k \in \mathbb{N}_0$ множество $\bigcup_{p \in V} \{Act(E, \mathfrak{A}) \mid \theta_L(E, i) \leq k\}$ является (а) сечением, и (б) согласованным сечением для любого вычисления E любого алгоритма $\mathfrak{A}$? Ответ обосновать

Значимые снимки

В качестве результата вычисления алгоритма сохранения снимка *хотелось бы* видеть снимок, представляющий собой конфигурацию заданного вычисления E

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 порядка $\prec$

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 в классе $\mathcal{E}$ и для вычисления E, если существует вычисление $E' \in \mathcal{E}$, в которое входит γ

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

1. **Снимок γ 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 причинно-следственно предшествующее ему действие β, 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 осуществимо)
   $m \in sent_{p \to q}^{\gamma} \quad \Rightarrow \quad$ (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 порядок $\preceq$ 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 $\mathcal{A}'$ действий $\mathcal{A} = Act(E, \mathfrak{A})$:

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

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

Осталось показать, что перестановка $\mathcal{A}'$ сохраняет порядок $\prec$

Задача 3: а попробуйте это сделать

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 $\prec$ и содержащее снимок γ

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, запускающийся параллельно с 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 и действия контрольными) и
2. алгоритма, над которым он надстраивается
   - (будем называть этот алгоритм, его сообщения и действия базовыми)