

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА
ВЕЛИКОГО

ФИЗИКО-МЕХАНИЧЕСКИЙ ИНСТИТУТ

ВЫСШАЯ ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ
ФИЗИКИ

Компьютерные сети
Отчёт по лабораторной работе №2
“Задача византийских генералов”

Выполнил:

Студент: Дамаскинский Константин

Группа: 5040102/10201

Принял:

к. ф.-м. н., доцент

Баженов Александр Николаевич

2022 г.

Содержание

| | |
|--------------------------------|---|
| 1. Постановка задачи | 2 |
| 2. Теория | 2 |
| 3. Реализация | 2 |
| 4. Результаты | 3 |
| 5. Обсуждение | 4 |
| 6. Приложения | 5 |

1. Постановка задачи

Пусть дано n генералов, m из которых византийские. Каждый генерал в начале располагает неким значением v_i , не известным другим генералам. Требуется разработать протокол взаимодействия, в результате следования которому каждый невизантийский генерал сформирует набор значений $\{u_i\}, i = \overline{1, n}$. Сформированный набор значений должен совпадать у всех генералов, при этом для индексов i , соответствующих невизантийским генералам, u_i должно совпадать с v_i . Будем считать, что каналы связи являются надёжными, а сообщения невозможно подделать. Необходимо реализовать алгоритм Лампорта-Шостака-Пиза для решения задачи византийских генералов.

2. Теория

Генералы будут общаться по протоколу, соответствующему частному случаю алгоритма Лампорта-Шостака-Пиза. Обмен сообщениями будет происходить в 2 этапа:

- На первом этапе каждый генерал передаёт всем остальным одно значение, при этом невизантийские генералы честно передают своё значение v_i , а византийские могут передавать произвольное значение (при этом он может передавать разным генералам разные значения). В результате у каждого генерала образуется вектор значений, пришедших ему от остальных
- На втором этапе каждый невизантийский генерал передаёт всем остальным вектор значений, сформированный на первом этапе, а византийский – вектор произвольных значений (потенциально различных для различных генералов)

В результате у каждого генерала формируется матрица информации, состоящая из вектора, сформированного на первом этапе, и векторов, полученных на втором этапе. Таким образом у генерала про каждого союзника формируется набор из нескольких (потенциально различных) значений. В качестве итогового значения, генерал выбирает наиболее часто встречающееся в наборе. Если таких значений несколько, то итоговое значение считается неопределённым. Алгоритм Лампорта-Шостака-Пиза гарантирует, что следуя его протоколу генералы всегда смогут прийти к консенсусу, в случае если $n > 3m$.

3. Реализация

Модель реализована на языке программирования Python. Все генералы работают в отдельных потоках, создаваемых с использованием модуля `threading`. Также в отдельных потоках работают все 3 канала связи между генералами. Для обеспечения потокобезопасности каналов используются `mutex` (класс `Lock`

из модуля threading). При переходе к следующему этапу алгоритма установлены точки барьерной синхронизации для всех генералов (класс Barrier из модуля threading).

На канальном уровне генералы общаются с помощью протокола SRP. Сетевой уровень для данной задачи тривиален, так как по условию предполагается, что канал связи существует между любой парой генералов.

4. Результаты

Рассмотрим пример работы алгоритма на модельном случае с $n = 5$, $m = 1$. В качестве индексов сопоставим генералам числа от 0 до 4 включительно. Последний генерал будет византийским, остальные – честными. Честным генералам изначально сопоставим значения вида t_i , где i – индекс генерала. Византийский генерал будет на первом этапе отправлять значения вида $m3_i$, где i – индекс генерала, которому адресовано сообщение, а на втором шаге – $m3_{ij}$, где i – индекс генерала, которому адресовано сообщение, j – индекс генерала, от которого (как утверждает византийский генерал) было получено это значение на первом этапе.

По результатам первого этапа генералами были сформированы следующие вектора:

```
General4 got: {0: 't0', 1: 't1', 2: 't2', 3: 't3'}
General0 got: {1: 't1', 2: 't2', 3: 't3', 4: 'm4_0'}
General3 got: {0: 't0', 1: 't1', 2: 't2', 4: 'm4_3'}
General2 got: {0: 't0', 1: 't1', 3: 't3', 4: 'm4_2'}
General1 got: {0: 't0', 2: 't2', 3: 't3', 4: 'm4_1'}
```

Рис. 1. $n = 5, m = 1$. Первый этап

На втором шаге получены следующие параметры:

```
General1 got: {0: {1: 't1', 2: 't2', 3: 't3', 4: 'm4_0'}, 2: {0: 't0', 1: 't1', 3: 't3', 4: 'm4_2'}, 3: {0: 't0', 1: 't1', 2: 't2', 4: 'm4_3'}, 4: {0: 't4_10', 1: 't4_11', 2: 't4_12', 3: 't4_13'}}
General0 got: {1: {0: 't0', 2: 't2', 3: 't3', 4: 'm4_1'}, 2: {0: 't0', 1: 't1', 3: 't3', 4: 'm4_2'}, 3: {0: 't0', 1: 't1', 2: 't2', 4: 'm4_3'}, 4: {0: 't4_00', 1: 't4_01', 2: 't4_02', 3: 't4_03'}}
General4 got: {0: {1: 't1', 2: 't2', 3: 't3', 4: 'm4_0'}, 1: {0: 't0', 2: 't2', 3: 't3', 4: 'm4_1'}, 2: {0: 't0', 1: 't1', 3: 't3', 4: 'm4_2'}, 3: {0: 't0', 1: 't1', 2: 't2', 4: 'm4_3'}}
General3 got: {0: {1: 't1', 2: 't2', 3: 't3', 4: 'm4_0'}, 1: {0: 't0', 2: 't2', 3: 't3', 4: 'm4_1'}, 2: {0: 't0', 1: 't1', 3: 't3', 4: 'm4_2'}, 4: {0: 't4_30', 1: 't4_31', 2: 't4_32', 3: 't4_33'}}
General2 got: {0: {1: 't1', 2: 't2', 3: 't3', 4: 'm4_0'}, 1: {0: 't0', 2: 't2', 3: 't3', 4: 'm4_1'}, 3: {0: 't0', 1: 't1', 2: 't2', 4: 'm4_3'}, 4: {0: 't4_20', 1: 't4_21', 2: 't4_22', 3: 't4_23'}}
```

Рис. 2. $n = 5, m = 1$. Второй этап

Затем путём выбора наиболее часто встречающегося элемента, генералы сформировали следующие результаты:

```
General4 result: {0: 't0', 1: 't1', 2: 't2', 3: 't3', 4: None}
General2 result: {0: 't0', 1: 't1', 2: 't2', 3: 't3', 4: None}
General0 result: {0: 't0', 1: 't1', 2: 't2', 3: 't3', 4: None}
General1 result: {0: 't0', 1: 't1', 2: 't2', 3: 't3', 4: None}
General3 result: {0: 't0', 1: 't1', 2: 't2', 3: 't3', 4: None}
```

Рис. 3. $n = 5, m = 1$. Третий этап

Можно заметить, что результаты у всех честных генералов совпадают, а также значения, полученные для честных генералов, соответствуют их реальным значениям (для византийского генерала значение в итоге оказалось неопределённым, так как на первом этапе он всем генералам рассылал разные значения). Можем сделать вывод, что задача византийских генералов решена корректно.

Тем не менее, у алгоритма есть ограничения. Например, если рассмотреть аналогичный случай при $n = 3$ и $m = 1$, византийским опять будет последний генерал, с индексом 2.

```
General1 got: {0: 't0', 2: 'm2_1'}
General0 got: {1: 't1', 2: 'm2_0'}
General2 got: {0: 't0', 1: 't1'}
General1 got: {0: {1: 't1', 2: 'm2_0'}, 2: {0: 't2_10', 1: 't2_11'}}
General2 got: {0: {1: 't1', 2: 'm2_0'}, 1: {0: 't0', 2: 'm2_1'}}
General0 got: {1: {0: 't0', 2: 'm2_1'}, 2: {0: 't2_00', 1: 't2_01'}}
General0 result: {0: None, 1: None, 2: None}
General2 result: {0: 't0', 1: 't1', 2: None}
General1 result: {0: None, 1: None, 2: None}
```

Рис. 4. $n = 3, m = 1$

Честным генералам удалось достичь формального консенсуса, так как их результирующие вектора совпадают (только при условии, что они “забывают” своё собственное значение, и пытаются восстановить его, действуя по протоколу), но при этом получить достоверную информацию о значениях друг друга честным генералам не удалось.

5. Обсуждение

В результате работы реализован алгоритм Лампорта-Шостака-Пиза для решения частного случая задачи Византийских генералов. Показана работоспособность алгоритма для $n = 5$ честных генералов и $m = 1$ византийского генерала среди них. Реализована модель взаимодействия между генералами (независимыми узлами) на сетевом и канальном уровне. Для обеспечения корректной

работы параллельного алгоритма были использованы различные примитивы синхронизации.

6. Приложения

1. Репозиторий с кодом программы и кодом отчёта:

<https://github.com/kystyn/networks>