

Санкт-Петербургский политехнический университет Петра Великого
Высшая школа прикладной математики и вычислительной физики
Кафедра прикладной математики

Лабораторная работа
по дисциплине «Компьютерные сети» на тему
«Задача Византийских генералов»

Выполнил

студент группы 5040102/10201

Д.А. Гусаров

Руководитель

доцент, к.ф.-м.н.

А.Н. Баженов

Санкт-Петербург

2022

Постановка задачи

Задачу Византийских генералов можно сформулировать так:

- Есть n генералов, f 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 , $i=1, \dots, n$.
- 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.

Реализация

Код проекта выложен на GitHub:

https://github.com/dimerf99/computer_networks/tree/main/lab_3

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.

Результаты

Рассмотрим пример работы алгоритма на модельном случае с $n = 5$ и $f = 1$:

- В качестве индексов сопоставим генералам числа от 0 до 4 включительно.
- Последний генерал будет византийским, остальные – честными.
- Честным генералам изначально сопоставим значения вида ti , где i – индекс генерала.
- Византийский генерал будет на первом этапе отправлять значения вида $f3_i$, где i – индекс генерала, которому адресовано сообщение, а на втором шаге – $f3_{ij}$, где i – индекс генерала, которому адресовано сообщение, j – индекс генерала, от которого (как утверждает византийский генерал) было получено это значение на первом этапе.

Вектора, сформированные на первом этапе:

Generals3 got: {0: 't0', 1: 't1', 2: 't2', 4: 'f4_3'}

Generals1 got: {0: 't0', 2: 't2', 3: 't3', 4: 'f4_1'}

Generals4 got: {0: 't0', 1: 't1', 2: 't2', 3: 't3'}

Generals2 got: {0: 't0', 1: 't1', 3: 't3', 4: 'f4_2'}

Generals0 got: {1: 't1', 2: 't2', 3: 't3', 4: 'f4_0'}

Вектора, сформированные на втором этапе:

```
General0 got: {1: {0: 't0', 2: 't2', 3: 't3', 4: 'f4_1'}, 2: {0: 't0', 1: 't1', 3: 't3', 4: 'f4_2'},  
3: {0: 't0', 1: 't1', 2: 't2', 4: 'f4_3'}, 4: {0: 'f4_00', 1: 'f4_01', 2: 'f4_02', 3: 'f4_03'}}  
General4 got: {0: {1: 't1', 2: 't2', 3: 't3', 4: 'f4_0'}, 1: {0: 't0', 2: 't2', 3: 't3', 4: 'f4_1'},  
2: {0: 't0', 1: 't1', 3: 't3', 4: 'f4_2'}, 3: {0: 't0', 1: 't1', 2: 't2', 4: 'f4_3'}}  
General2 got: {0: {1: 't1', 2: 't2', 3: 't3', 4: 'f4_0'}, 1: {0: 't0', 2: 't2', 3: 't3', 4: 'f4_1'},  
3: {0: 't0', 1: 't1', 2: 't2', 4: 'f4_3'}, 4: {0: 'f4_20', 1: 'f4_21', 2: 'f4_22', 3: 'f4_23'}}  
General3 got: {0: {1: 't1', 2: 't2', 3: 't3', 4: 'f4_0'}, 1: {0: 't0', 2: 't2', 3: 't3', 4: 'f4_1'},  
2: {0: 't0', 1: 't1', 3: 't3', 4: 'f4_2'}, 4: {0: 'f4_30', 1: 'f4_31', 2: 'f4_32', 3: 'f4_33'}}  
General1 got: {0: {1: 't1', 2: 't2', 3: 't3', 4: 'f4_0'}, 2: {0: 't0', 1: 't1', 3: 't3', 4: 'f4_2'},  
3: {0: 't0', 1: 't1', 2: 't2', 4: 'f4_3'}, 4: {0: 'f4_10', 1: 'f4_11', 2: 'f4_12', 3: 'f4_13'}}
```

Вектора, сформированные путём выбора наиболее часто встречающегося элемента:

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

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 рассылал всем генералам разные значения).

Вывод – задача Византийских генералов решена корректно.

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

```
General1 got: {0: 't0', 2: 'f2_1'}
General0 got: {1: 't1', 2: 'f2_0'}
General2 got: {0: 't0', 1: 't1'}
General2 got: {0: {1: 't1', 2: 'f2_0'}, 1: {0: 't0', 2: 'f2_1'}}
General1 got: {0: {1: 't1', 2: 'f2_0'}, 2: {0: 'f2_10', 1: 'f2_11'}}
General0 got: {1: {0: 't0', 2: 'f2_1'}, 2: {0: 'f2_00', 1: 'f2_01'}}
General0 result: {0: None, 1: None, 2: None}
General1 result: {0: None, 1: None, 2: None}
General2 result: {0: 't0', 1: 't1', 2: None}
```

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.

Выводы

В результате работы реализован алгоритм Лампорта-Шостака-Пиза для решения частного случая задачи Византийских генералов. Показана работоспособность алгоритма для $n = 5$ честных генералов и $f = 1$ 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.

Использованная литература

1. А.Н. Баженов, Компьютерные сети, курс лекций