

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

Лабораторная работа
по дисциплине «Компьютерные сети»
на тему
**Реализация алгоритма Лампорта для задачи
Византийских генералов**

Выполнил

студент гр. 5040102/10201

Долгий В.С.

/_____/

Руководитель

доцент, к.ф.-м.н.

Баженов А.Н.

/_____/

Постановка задачи

Задачу Византийских генералов сформулируем в следующем виде [2]:

Имеется n генералов, f из которых – византийские. Каждый генерал в начале располагает неким значением V_i , неизвестной другим генералам. Требуется разработать протокол взаимодействия, в результате следования которому каждый невинантйский генерал сформирует набор значений \tilde{V} , $i = 1, \dots, n$. Сформированный набор значений должен совпадать у всех генералов, при этом для индексов i , соответствующих невинантйским генералам, \tilde{V} должно совпадать с V_i .

При этом будем считать, что каналы связи являются надёжными, а сообщения невозможно подделать.

Необходимо реализовать алгоритм Лампорта-Шостака-Пиза для решения задачи Византийских генералов.

Теория

Генералы будут общаться по протоколу, соответствующему частному случаю алгоритма Лампорта-Шостака-Пиза [3]. Обмен сообщениями будет происходить в 2 этапа:

- На первом этапе каждый генерал передаёт всем остальным одно значение, при этом невинантйские 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 невинантйский 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 $n > 3f$.

Реализация

Модель реализована на языке Python. Все генералы работают в отдельных потоках, создаваемых с использованием модуля `threading`. Также в отдельных потоках работают все

каналы связи между генералами. Для обеспечения потокобезопасности каналов используются `mutex`'ы (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 [1], реализованного в первой лабораторной. Сетевой уровень для данной задачи тривиален, так как по условию предполагается, что канал связи существует между любой парой генералов.

Код проекта выложен на GitHub:

<https://github.com/Ragnarok7861/CompNET>

Результаты

Рассмотрим пример работы алгоритма на модельном случае с $n = 4$ и $f = 1$.

В качестве индексов сопоставим генералам числа от 0 до 3 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.

По результатам первого этапа генералами были сформированы следующие вектора:

```
General0 got: {1: 't1', 2: 't2', 3: 'f3_0'}
General1 got: {0: 't0', 2: 't2', 3: 'f3_1'}
General2 got: {0: 't0', 1: 't1', 3: 'f3_2'}
General3 got: {0: 't0', 1: 't1', 2: 't2'}
```

Во время второго шага, генералы получили такой набор векторов:

```
General0 got: {1: {0: 't0', 2: 't2', 3: 'f3_1'},
               2: {0: 't0', 1: 't1', 3: 'f3_2'},
               3: {0: 'f3_00', 1: 'f3_01', 2: 'f3_02'}}
General1 got: {0: {1: 't1', 2: 't2', 3: 'f3_0'},
               2: {0: 't0', 1: 't1', 3: 'f3_2'},
               3: {0: 'f3_10', 1: 'f3_11', 2: 'f3_12'}}
General2 got: {0: {1: 't1', 2: 't2', 3: 'f3_0'},
               1: {0: 't0', 2: 't2', 3: 'f3_1'},
               3: {0: 'f3_20', 1: 'f3_21', 2: 'f3_22'}}
General3 got: {0: {1: 't1', 2: 't2', 3: 'f3_0'},
               1: {0: 't0', 2: 't2', 3: 'f3_1'},
               2: {0: 't0', 1: 't1', 3: 'f3_2'}}
```

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

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

Как видим, 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 $f = 1$. Византийским опять будет последний генерал, с индексом 2.

```
General0 got: {1: 't1', 2: 'f2_0'}
General1 got: {0: 't0', 2: 'f2_1'}
General2 got: {0: 't0', 1: 't1'}

General1 got: {0: {1: 't1', 2: 'f2_0'}, 2: {0: 'f2_10', 1: 'f2_11'}}
General2 got: {0: {1: 't1', 2: 'f2_0'}, 1: {0: 't0', 2: 'f2_1'}}
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 о значениях друг друга честным генералам не удалось.

Выводы

В рамках был реализован алгоритм Лампорта-Шостака-Пиза для решения частного случая задачи Византийских генералов. Реализация опирается на канальный и сетевой уровни, разработанные в предыдущих лабораторных. Для обеспечения корректной работы параллельного алгоритма были использованы различные примитивы синхронизации.

Использованная литература

1. А.Н. Баженов, Компьютерные сети, курс лекций
2. neerc.ifmo.ru, Проблема византийских генералов, электронный ресурс
https://neerc.ifmo.ru/wiki/index.php?title=Проблема_византийских_генералов
3. neerc.ifmo.ru, Алгоритм Лампорта-Шостака-Пиза, электронный ресурс
https://neerc.ifmo.ru/wiki/index.php?title=Алгоритм_Лампорта-Шостака-Пиза