

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное  
учреждение высшего образования  
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук  
имени И. И. Воровича

Направление подготовки  
01.03.02 — Прикладная математика  
и информатика

РЕАЛИЗАЦИЯ ОТКАЗОУСТОЙЧИВОГО ПРОСТРАНСТВА КОРТЕЖЕЙ  
СРЕДСТВАМИ ЯЗЫКА PYTHON

Выпускная квалификационная работа  
на степень бакалавра

Студентки 4 курса  
В. М. Ложкиной

Научный руководитель:  
старший преподаватель кафедры информатики и вычислительного  
эксперимента В. Н. Брагилевский

Допущено к защите:

руководитель направления ФИИТ \_\_\_\_\_ В. С. Пилиди

Ростов-на-Дону  
2018

# Содержание

Введение . . . . .	3
1. Теоретические основы . . . . .	4
1.1. Распределённые вычислительные системы . . . . .	4
1.2. Задача византийских генералов . . . . .	5
1.3. Paxos . . . . .	7
1.4. Пространство corteжей . . . . .	8
1.5. Византийское пространство corteжей . . . . .	9

## Введение

В современном мире распределённые системы являются основой различных объектов, например, веб-сервисов и пиринговых сетей. Поскольку обмен сообщениями в системе осуществляется по ненадёжным каналам связи, сообщения, посылаемые системе или самой системой, могут просматриваться, перехватываться и подменяться. Это приводит к неправильному функционированию системы, отказу отдельных компонент или системы в целом. Поэтому в распределённых системах поддержке безопасности уделяется особое внимание.

Для повышения надёжности системы случайные и умышленные сбои в системе интерпретируются как Византийские ошибки (в терминах «Задачи о Византийских генералах»). В этом случае можно использовать соответствующие отказоустойчивые техники, которые защитят систему и от случайных, и от умышленных сбоев.

В данной работе рассматривается Византийское пространство кортежей — открытая распределённая устойчивая к Византийским ошибкам система, основанная на пространстве кортежей без распределённой памяти, в которой процессы взаимодействуют путём обмена сообщениями.

# 1. Теоретические основы

## 1.1. Распределённые вычислительные системы

Распределённая вычислительная система — это набор независимых компьютеров, реализующий параллельную обработку данных на многих вычислительных узлах. С точки зрения пользователя этот набор является единым механизмом, предоставляющим полный доступ к ресурсам. Существует возможность добавления новых ресурсов, свойств и методов, возможность перераспределения ресурсов по системе, но информация об этих событиях скрыта от пользователя [Tanenbaum].

Распределённая система, рассматриваемая в данной работе, представлена множеством из  $n$  серверов. Взаимодействие клиентов с системой происходит с помощью вспомогательной промежуточной инфраструктуры, как показано на рисунке 1.

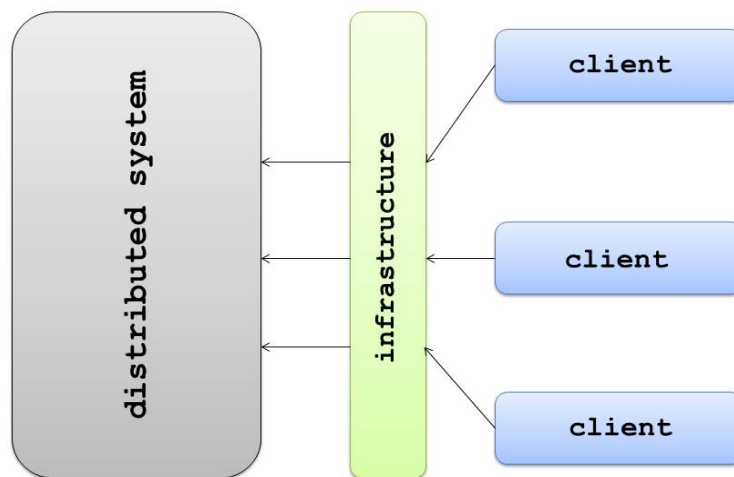


Рисунок 1 — Взаимодействие клиентов с распределённой системой

Одной из важнейших характеристик распределённых систем является отказоустойчивость. Отказоустойчивость — это свойство си-

системы сохранять работоспособность в том случае, если какие-либо составляющие её компоненты перестали правильно функционировать. Компоненты системы могут стать неработоспособны по различным причинам, например, из-за технологических сбоев или атак безопасности.

Большинство современных распределённых систем имеют характеристики открытых систем. Открытая распределённая система предполагает использование служб, вызов которых требует стандартного синтаксиса и семантики. Такая система может иметь неизвестное количество ненадёжных и неоднородных участников, то есть участникам не нужно быть активными одновременно (свойство разъединённости во времени) и не обязательно знать друг о друге (свойство разъединённости в пространстве). Связь между узлами распределённой системы является ненадёжной (может прерываться, что повлечёт за собой потерю сообщений), обмен сообщениями может происходить не мгновенно, а с существенной задержкой. Кроме того, любой узел системы может отказать или быть выключен в любой момент времени. Все эти факторы неизбежно приводят к неправильному функционированию системы. Один из способов улучшить её надёжность — это интерпретировать случайные или умышленные неполадки как Византийские ошибки (в терминах «Задачи византийских генералов»), тогда использование отказоустойчивых техник сможет сделать координационную составляющую системы отказоустойчивой и для случайных сбоев, и для умышленных вторжений.

## **1.2. Задача византийских генералов**

Задача византийских генералов — это задача взаимодействия нескольких удалённых абонентов, получивших сообщения из одного центра, причём часть этих абонентов, в том числе центр, могут быть предателями, то есть могут посылать заведомо ложные сообщения с целью дезинформирования. Нахождение решения задачи заключа-

ется в выработке единой стратегии действий, которая будет являться выигрышной для всех абонентов.

Формулировка задачи состоит в следующем. Византийская армия представляет собой объединение некоторого числа легионов, каждым из которых командует свой генерал, генералы подчиняются главнокомандующему армии Византии. Поскольку империя находится в упадке, любой из генералов и даже главнокомандующий могут быть заинтересованы в поражении армии, то есть являться предателями. Генералов, не заинтересованных в поражении армии, будем называть верными. В ночь перед сражением каждый из генералов получает от главнокомандующего приказ о действиях во время сражения: атаковать или отступить. Таким образом, имеем три возможных исхода сражения:

- Благоприятный исход: все генералы атакуют противника, что приведёт к его уничтожению и победе Византии.
- Промежуточный исход: все генералы отступят, тогда противник не будет побеждён, но Византия сохранит свою армию.
- Неблагоприятный исход: некоторые генералы атакуют противника, некоторые отступят, тогда Византийская армия потерпит поражение.

Так как главнокомандующий тоже может оказаться предателем, то генералам не следует доверять его приказам. Однако если каждый генерал будет действовать самостоятельно, независимо от других генералов, то вероятность наступления благоприятного исхода становится низкой. Таким образом, генералам следует обмениваться информацией между собой для того, чтобы прийти к единому решению.

На практике задача византийских генералов решается с помощью алгоритмов консенсуса, ярким представителем которых является алгоритм Паксос, предложенный Лесли Лампортом.

### 1.3. Paxos

Ракос — это алгоритм для решения задачи консенсуса в сети ненадёжных вычислителей. Компоненты распределённой системы можно разделить на 3 группы:

- Заявитель (Proposer) — выдвигает «предложения» (какие-либо значения), которые либо принимаются, либо отвергаются в результате работы алгоритма консенсуса.
- Акцептор (Acceptor) — принимает или отвергает «предложение» Заявителя, согласует своё решение с остальными Акцепторами, уведомляет о своём решении Узнающих. Если Акцепторами было принято какое-либо значение, предложенное Заявителем, то оно называется утверждённым.
- Узнающий (Learner) — запоминает решения Акцепторов, принятые в результате работы алгоритма консенсуса.

Компоненты распределённой системы могут принадлежать сразу нескольким группам, описанным выше, и вести себя и как Заявитель, и как Акцептор, и как Узнающий.

Такое распределение ролей в системе гарантирует следующее:

- Только предложенное Заявителем значение может быть утверждено Акцепторами.
- Акцепторами утверждается только одно значение из всех предложенных Заявителями значений (возможно, противоречивых).
- Узнающий не сможет узнать об утверждении какого-либо значения вплоть до того момента, пока оно действительно не будет утверждено.

## 1.4. Пространство кортежей

Кортеж — это структура данных, представляющая собой неизменяемый список фиксированной длины, элементы которого могут относиться к различным типам данных. Два кортежа считаются равными, если совпадают их длины, а также типы и значения соответствующих полей.

Хранилище кортежей, в котором доступ к элементам может осуществляться параллельно, называется пространством кортежей. Оно является основой языка программирования Linda. Данный язык предназначен для построения эффективных параллельных программ. Он включает в себя три операции манипулирования данными (кортежами):

- out - запись кортежа в пространство кортежей.
- rd - недеструктивное чтение кортежа.
- in - деструктивное чтение (извлечение) кортежа.

Шаблон кортежа будем называть кортеж, некоторые поля которого неопределены и не представляют важности. Кортеж соответствует шаблону, если длина кортежа равна длине шаблона и определённые в шаблоне поля совпадают по типу и значению с соответствующими полями кортежа. Операция записи out принимает в качестве входного параметра кортеж, все поля которого определены. Операции чтения rd и in принимают в качестве входного параметра шаблон кортежа, по которому производится поиск соответствующих кортежей в пространстве. Таким образом, пространство кортежей можно рассматривать как разновидность распределённой памяти: например, одна группа процессов записывает данные в пространство кортежей, а другая группа процессов извлекает данные из пространства и использует их в своей дальнейшей работе.

Для того, чтобы поиск кортежей в пространстве кортежей занимал минимальное время, адресация в нём осуществляется по содер-



жимому (например, с помощью алгоритмов хэширования). Иными словами, пространство кортежей — реализация парадигмы ассоциативной памяти.

В данной работе рассматривается распределённая система под названием «Византийское пространство кортежей». Основой системы является пространство кортежей, имеющее ряд особенностей, которые позволяют добиться отказоустойчивости.

### **1.5. Византийское пространство кортежей**