

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

01.03.02 Прикладная математика и информатика

(код и наименование направления подготовки, специальности)

Системное программирование и компьютерные технологии

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему Способы моделирования, анализа и реализации децентрализованных
пиринговых систем

Студент

Ф.М. Бабаева

(И.О. Фамилия)

(личная подпись)

Руководитель

А.В. Очеповский

(И.О. Фамилия)

(личная подпись)

Консультанты

М. А. Четаева

(И.О. Фамилия)

(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« _____ » _____ 20 _____ г.

Тольятти 2018

Аннотация

Тема бакалаврской работы – «Способы моделирования, анализа и реализации децентрализованных пиринговых систем».

Цель работы: описать модели полностью децентрализованных систем и разработать программное обеспечение для анализа данных систем.

Объект исследования: процесс функционирования пиринговых систем.

Предмет исследования: модели и алгоритмы исследования распределенных пиринговых систем.

Работа посвящена исследованию важной области в современной информатике – изучению принципов построения и функционирования распределенных систем и алгоритмов на примере распределенных пиринговых систем. Данный вид виртуальной сетевой топологии активно исследуется и применяется при построении групповых интеллектуальных систем таких как самонастраивающиеся сети, управление роем объектов, распределенные реестры, вычисления и хранилища. Поэтому описание моделей распределенных пиринговых систем и программного обеспечения для их анализа является актуальной задачей.

В первой главе рассматривается классификация и свойства пиринговых систем, а также приводится список общих алгоритмов, которые должны поддерживать P2P сети.

Во второй главе проводится обзор методов формального моделирования, языков программирования и симуляторов для компьютерного имитационного моделирования, реализации и исследования пиринговых систем.

Результатами работы является научный обзор литературы в области теории пиринговых систем, методов формального моделирования P2P сетей, специализированных языков программирования для реализации распределенных вычислений и алгоритмов, P2P симуляторов.

Пояснительная записка выполнена на 40 страницах, включает 5 рисунков. Список используемых источников включает 54 источника, из них 37 на иностранном языке.

Abstract

The theme of the bachelor's work is "Models of decentralized peer-to-peer systems".

Objective: to describe models of fully decentralized systems and to develop software for the analysis of these systems.

Object of research: the process of functioning of peer-to-peer systems.

The subject of the study: models and algorithms for investigating distributed peer-to-peer systems.

The work is devoted to the study of an important area in modern computer science - the study of the principles of construction and operation of distributed systems and algorithms using the example of distributed peer-to-peer systems. This type of virtual network topology is actively explored and used in the construction of group intelligent systems such as self-tuning networks, control of the object swarm, distributed registries, calculations and storages. Therefore, the description of models of distributed peer-to-peer systems and software for their analysis is an urgent task.

The first chapter looks at the classification and properties of peer-to-peer systems, and provides a list of common algorithms that should support P2P networks.

The second chapter reviews the methods of formal modeling, programming languages and simulators for computer simulation, implementation and research of peer-to-peer systems.

The result of this work is a scientific review of the literature in the field of the theory of peer-to-peer systems, methods of formal modeling of P2P networks, specialized programming languages for implementing distributed computations and algorithms, P2P simulators.

Bachelor's work consists of introduction, three chapters, conclusion, list of used sources and applications. The explanatory note is made on 40 pages, includes 5 drawings tables. The list of sources includes 54 sources, of which 37 in a foreign language.

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ | 5 |
| 1 Основные понятия теории пиринговых систем | 8 |
| 1.1 Понятие пиринговой сети..... | 8 |
| 1.2 Классификация P2P систем..... | 11 |
| 1.3 Типы оверлеев пиринговых сетей | 15 |
| 1.3.1 Неструктурированные оверлеи..... | 15 |
| 1.3.2 Структурированные оверлеи | 17 |
| 1.4 Основные распределенные алгоритмы децентрализованных пиринговых систем..... | 18 |
| 1.4.1 Основные принципы и проблемы разработки распределенных систем | 18 |
| 1.4.2 Обобщенный список распределенных алгоритмов децентрализованных пиринговых систем..... | 20 |
| 2 Моделирование децентрализованных пиринговых систем | 23 |
| 2.1 Методы формального моделирования распределенных вычислений и систем | 23 |
| 2.2 Языки программирования для распределенных и мобильных вычислений | 25 |
| 2.2.1 Язык программирования Pict..... | 25 |
| 2.2.2 Язык программирования SALSА..... | 26 |
| 2.2.3 Язык программирования JoCalm | 27 |
| 2.3 Симуляторы для исследования пиринговых систем | 28 |
| 2.3.1 Симулятор PeerSim | 29 |
| 2.3.2 Симулятор GPS..... | 30 |
| 2.3.3 Симулятор NeuroGrid | 30 |
| 2.3.4 Симулятор P2PSim..... | 31 |
| ЗАКЛЮЧЕНИЕ | 33 |
| СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ..... | 35 |

ВВЕДЕНИЕ

Информационные системы и технологии пронизывают все новые и новые сферы человеческой деятельности, за которыми стоят решения все более сложных научных и практических задач. Постоянно возрастают требования к скорости передачи, обработки и анализа данных, что вызывает соответствующие требования к скорости вычислительных процессоров и компьютерных сетей, объему хранимой информации, надежности систем передачи данных и т.п. Технологический прогресс в области цифровой схемотехники делает возможность повсеместного использования миниатюрных вычислительных систем и их связи с классическими архитектурами.

Несмотря на имеющиеся скептические прогнозы по поводу выполнимости закона Мура, характеристики процессоров, сетей и накопителей данных продолжают свой экспоненциальный рост. Действительно, проанализировав открытые данные можно убедиться, что за последние 40-45 лет указанные характеристики увеличились на шесть порядков [7].

Если такие темпы роста сохраняться, то к 2031 году мы сможем увидеть дальнейшее тысячекратное улучшение характеристик аппаратных средств. В результате чего мобильные устройства, ноутбуки и персональные компьютеры будут иметь более 1000 процессоров. Сетевые магистрали будут иметь скорость более 100 терабит в секунду, а емкость накопителей данных будет измеряться в петабайтах.

Технологии глобальной сети Интернет также активно развиваются. Одним из основных факторов, обуславливающих научные и инженерные поиски в области новых технологий Интернет является продолжающийся «взрывной» рост трафика, который «делает затруднительным его концентрацию на отдельных узлах сети, и как следствие, порождает уже явно выраженную в последнее время тенденцию к переходу различных интернет-сервисов от клиент-серверной модели предоставления услуг сначала к частично распределённой ..., а затем и к полностью распределённой» [39]. Анализ динамики трафика Интернет показывает, что сегодня доля трафика

традиционных сервисов (электронная почта, WWW, FTP) по сравнению с остальными видами трафика, достаточно мала и с течением времени только будет уменьшаться. Наибольший же рост объемов интернет трафика наблюдается для трафика P2P сервисов и приложений, доля которого уже в 2007 году достигла 70% [39].

Термин «P2P» используется для описания широкого круга приложений и архитектур. Часть пиринговых систем была успешно апробирована на системах распределенных вычислений (SETI@home, XtremWeb), распределенного хранения данных (Publius, Kazaa, Gnutella), обмена сообщений (ICQ, Jabber), массовых онлайн играх и т.д. Во всех этих приложениях, на узлах распределяется часть ресурсов (вычислительная мощность, дисковое пространство и т.д.), непосредственно доступных другим узлам, распределенным по сети.

В последнее время технология пиринговых сетей активно используется на рынке онлайн-телевидения – P2P-TV [40], что обеспечило высокий научный интерес к разработке новых технологий построения распределенных технологий и алгоритмов.

Вместе с тем информация, относящаяся к методам формального описания, математического и компьютерного имитационного моделирования пиринговых систем, ровно как, и распределенных алгоритмов, применяемых в P2P сетях, носит разрозненный, несистематический характер. Кроме того, наблюдается острый дефицит литературы на русском языке, посвященной теме исследований. Поэтому тема работы является **актуальной** как с научной, так и с инженерной точек зрения.

Объектом исследований являются методология пиринговых систем. **Предметом** исследований являются систематизация методов формального описания, моделирования и реализации децентрализованных пиринговых систем.

Целью бакалаврской работы является системный анализ подходов и методов формального описания, математического и компьютерного моделирования, а также реализации децентрализованных пиринговых систем.

Для достижения поставленной цели были выделены следующие задачи:

1. На основе анализа современной научной литературы обобщить основную информацию по теории пиринговых систем.
2. Выполнить обзор методов формального описания и моделирования пиринговых систем.
3. Выполнить обзор специализированных языков программирования, предназначенных для реализации распределенных систем и вычислений.
4. Выполнить обзор программ-симуляторов P2P сетей.

Новизна исследования заключается в системном изложении научных знаний в области формального описания, моделирования и реализации децентрализованных пиринговых сетей.

Бакалаврская работа состоит из введения, двух глав, заключения, списка используемых источников.

В первой главе систематизируются основные положения теории пиринговых систем и основные подходы к формальному описанию P2P сетей.

Во второй главе рассматриваются языки реализации и приводится обзор программ симуляторов пиринговых сетей.

1 ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ПИРИНГОВЫХ СИСТЕМ

1.1 Понятие пиринговой сети

Глобальные сетевые технологии прочно вошли в жизнь человечества и продолжают активно развиваться. При этом продолжает наблюдаться не только экспоненциальный рост общего трафика [35] сети Интернет, но и перераспределение его структуры. Доля традиционных коммуникаций таких как электронная почта, ftp и даже web продолжает сокращаться при одновременном росте трафика пиринговых и частично централизованных систем, таких как IoT, P2P-TV, мобильного видео трафика с применением технологий AR и VR [39].

Так в соответствии с отчетом Cisco «Наглядный индекс развития сетевых технологий» (Cisco Visual Networking Index™ Complete Forecast, Cisco VNI) [52], «глобальная цифровая трансформация ... по 2021 гг. будет по-прежнему существенно влиять на запросы и потребности IP-сетей». При этом число интернет-пользователей к 2021 году увеличится до 4,6 млрд. человек и составит 58% мирового населения. Также ускоренными темпами будут расти:

- количество межмашинных соединений и персональных устройств;
- средняя скорость широкополосного доступа;
- прирост видеотрафика.

В целом ожидается, что объем мирового IP-трафика к 2021 г. достигнет 3,3 зеттабайт, в то время как в 2017 данный показатель составлял немногим более 1,6 зеттабайт.

По данным сайта pag.ru [49], как и отмечалось выше, к 2021 году основную долю в Интернет-трафике будут составлять технологии мобильного видео, основу которых составляют P2P-TV сети. На рисунке 1.1 приводится прогноз трафика [49].

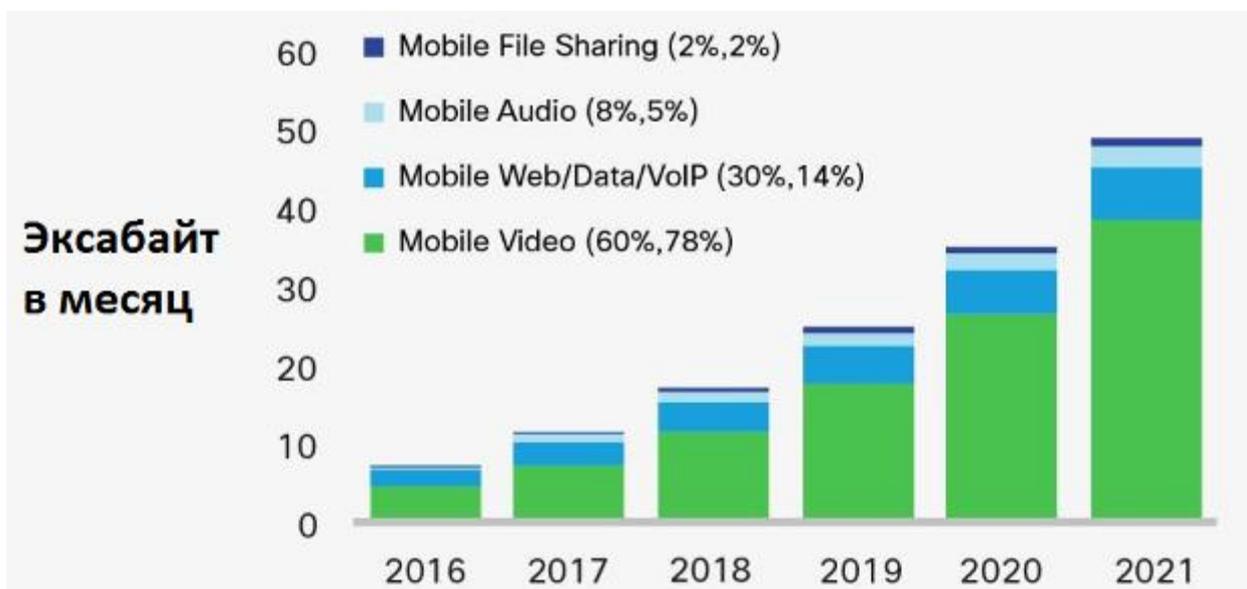


Рисунок 1.1 - Прогнозирование распределения трафика по видам сообщений

Таким образом доля трафика мобильного видео в ближайшей перспективе составит 80% от общего трафика сообщений глобальной сети.

Анализ литературы [40, 51] показывает, что основу перспективных высоконагруженных распределенных систем составляют и будут составлять децентрализованные распределенные системы, а именно пиринговые, P2P, системы и технологии. Как известно, на основе P2P технологий строятся системы распределённого реестра (blockchain, криптовалюты, умные контракты), системы распределенного телевидения P2P-TV, системы класса IoT, массовые онлайн игры (ММОГ) и многое другое.

Таким образом, мы видим, что пиринговые системы активно используются в современных информационных технологиях, а в перспективе ожидается рост внимания к совершенствованию механизмов построения и функционирования P2P систем.

Прежде чем дать понятие пиринговой системы, обсудим понятие распределенных систем, разновидностью которых являются пиринговые системы.

Несмотря на то, что в русскоязычной литературе под распределенными системами понимается, в основном, информационные системы с несколькими базами данных, по мнению автора такое определение не является полным.

В данной работе под распределенной системой мы будем понимать *множество вычислительных узлов, соединенных коммуникационной сетью и взаимодействующих для выполнения общих задач* [35]. Таким образом, из данного определения можно выделить следующие требования к распределенным системам:

- узлы распределенной системы должны быть независимыми, чтобы они могли действовать автономно;
- узлы должны соединяться посредством коммуникационной сети, чтобы каждый узел имел прямую или косвенную связь с другими узлами;
- должен иметься механизм координации деятельности узлов для решения поставленных задач.

Если привести классификацию распределенных информационных систем на основе архитектуры [7], то можно выделить следующие виды:

- модель «Клиент-сервер»;
- модель «Master-Slaves», как разновидность модели клиент-сервер;
- модель «точка-точка», peer-to-peer (P2P);

Модель «точка-точка», peer-to-peer (P2P) является основой для создания пиринговых распределенных систем. Ярким примером пиринговой системы являются торренты. На рисунке 1.2 показана архитектура пиринговой системы, из которого видно, что в этой архитектуре отсутствует узел, управляющий работой других узлов.

Механизм взаимодействия элементов в пиринговой системе основан на широковещательной или многоадресной рассылке, которая позволяет организовать обмен данными. Именно связь «точка-точка» оказывается эффективной, когда между элементами пиринговой системы передаются большие объемы данных.

P2P сети работают на абстрактных P2P протоколах. Если рассматривать стандартную модель стека сетевых протоколов TCP/IP, то P2P протоколы можно отнести к прикладному уровню. Таким образом, P2P сети работают, как

правило, поверх глобальной Интернет, являясь оверлейными (overlay, наложенными) сетями [42].

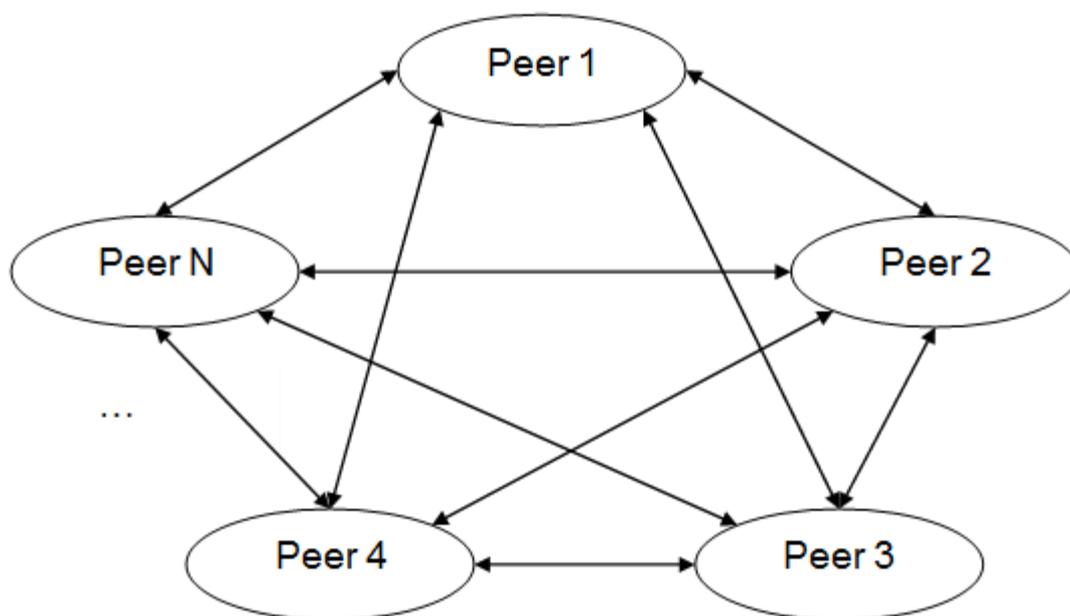


Рисунок 1.2 – Архитектура модели peer-to-peer

Пиринговые системы могут образовывать различные сетевые топологии: кольцо, звезда, дерево. Эти топологии позволяют обеспечить достижение следующих целей:

- определить постоянное положение элемента в системе;
- обеспечить маршрутизацию сообщений между элементами.

В таких конфигурациях каждый узел может выступать в роли сервера и одновременно клиента для другого узла.

1.2 Классификация P2P систем

P2P системы можно классифицировать по двум критериям: степени централизации и структуре [7].

По степени централизации архитектуры пиринговых систем можно разделить на:

- централизованные (Centralized);
- частично централизованные (Partially centralized);

– распределенные (Distributed).

Если пиринговая система строится по централизованной архитектуре, то в архитектуре системы выделяется группа узлов-серверов (рисунок 1.3 [7]), выполняющих критические операции такие как регистрация и определение положение узла в сети, а также распределение ресурсов по узлам. Как правило, узлы-серверы соединяются посредством LAN-технологий, что обеспечивает высокую скорость передачи информации между ними.

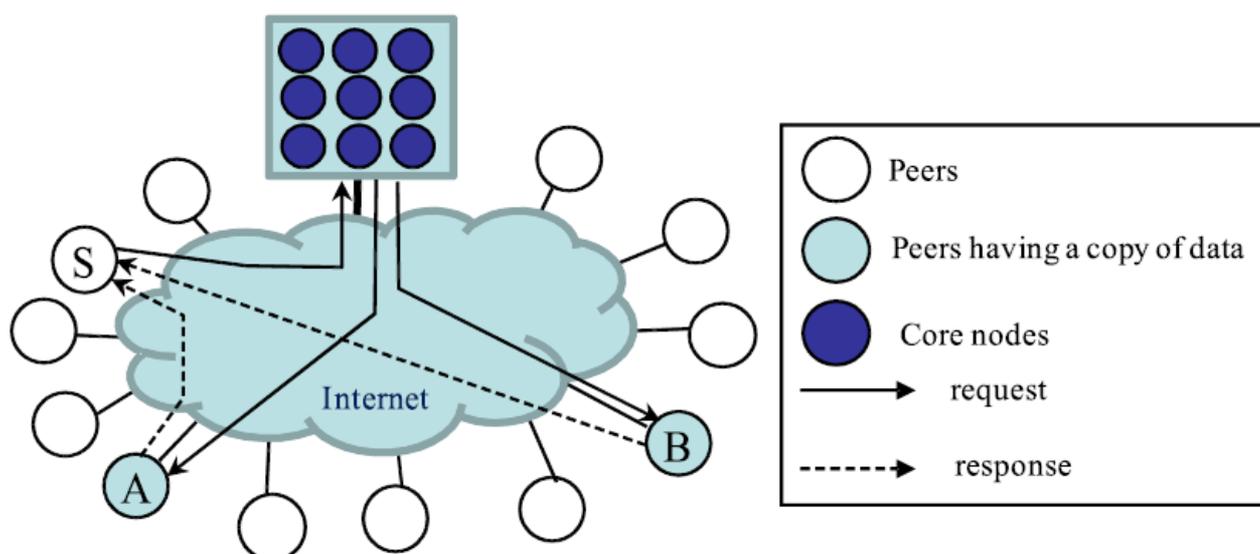


Рисунок 1.3 – Централизованная пиринговая сеть

Рисунок поясняет алгоритм определения ресурсов в централизованной пиринговой сети. Узел S запрашивает некоторые ресурсы, хранящиеся на пирах сети, через узлы-серверы. Серверные узлы, хранящие таблицы размещения ресурсов, оповещают некоторые узлы (A и B) о том, что узел S запросил данные, которые на них размещены. Кроме того, серверные узлы передают узлам A и B положение узла S в сети. После этого узлы A и B передают информацию узлу S.

Централизованная архитектура обладает рядом преимуществ, таких как:

- возможность организации и хранения метаданных обо всех ресурсах в сети;
- осуществление быстрых операций индексирования, упорядочивания, поиска ресурсов в сети;

- относительно небольшое количество поисковых запросов в сети;
- балансировка нагрузки сети;
- легкость управления относительно небольшим числом узлов-серверов.

Вместе с тем данная архитектура имеет ряд больших недостатков:

- центральные узлы являются единственной точкой отказа, поэтому выход из строя или потеря доступа к кластеру серверных узлов делает неработоспособной всю сеть;
- централизованное решение не является в полной мере масштабируемым, так как весь трафик проходит через единое место;
- к серверным узлам предъявляются высокие требования по памяти и быстродействию.

Примерами пиринговых систем, построенных по централизованному принципу, являются BitTorrent, Napster, OceanStore [7].

Для устранения недостатков централизованных пиринговых систем были разработаны частично централизованные пиринговые системы, отличающиеся от предыдущих систем тем, что теперь критические задачи выполняют несколько групп узлов-серверов. Если использовать двухуровневую иерархию, то в таких сетях выделяют группы супер-узлов – мощных узлов, соединённых высокоскоростными каналами связи, и групп простых узлов, соединённых с одной или несколькими группами суперузлов. На сегодняшний день двухуровневая иерархия является одной из наиболее часто используемой. Примерами таких систем являются Kazaa и eDonkey.

На рисунке 1.4 [7] представлена схема взаимодействия пиров в частично-структурированной сети. Узел S запрашивает ресурс у своего суперузла SP1. Суперузел SP1 просматривает свои данные о размещении искомого ресурса и инициализирует передачу данных с ресурса A, при этом узел B, также имеющий запрашиваемый ресурс, в передаче информации не используется (например, по причине загрузки или временного отсутствия доступа). Кроме того, суперузел SP1 передает поисковый запрос другим, связанным с ним,

суперузлам. В результате распространения запроса суперузел SP3 инициализирует передачу данных от узла C к узлу S.

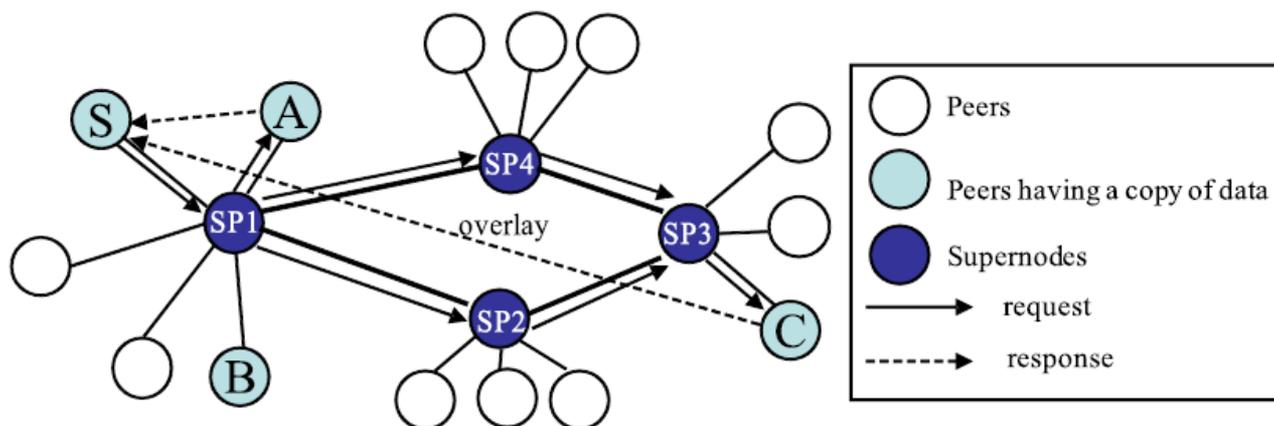


Рисунок 1.4 – Частично централизованная пиринговая сеть

Несмотря на то, что частично централизованные системы решают ряд проблем централизованных пиринговых систем, им также присущи недостатки, главными из которых являются сложности алгоритмов обработки метаданных, таких как индексация и поиск данных, а также учета присоединения и отсоединения узлов к сети. Кроме того, частично централизованные пиринговые системы оказываются неэффективными, когда число пиров в сети становится большим или идет интенсивный процесс присоединения-отсоединения пиров из сети.

В этих случаях используются децентрализованные пиринговые системы, которые не используют никакой иерархии пиров в сети. В этой архитектуре любой пир входит в состав глобального оверлея P2P сети и может хранить метаданные о ресурсах только своих близлежащих соседях. Кроме того, для устойчивости сети при отказе отдельного пира информация о метаданных реплицируется всем непосредственным соседям по оверлею.

Так как в таких сетях отсутствует единые точки отказа, то децентрализованные пиринговые сети могут быстро восстанавливаться после большого числа разнообразных сбоев.

Однако децентрализованные алгоритмы оказываются достаточно сложными и могут вовлекать в свое выполнение большое количество узлов, что делает их разработку и анализ сложной и инженерной и научной задачей.

В дальнейшем в работе будут рассматриваться децентрализованные пиринговые системы.

1.3 Типы оверлеев пиринговых сетей

Независимо от вида архитектуры пиринговых сетей узлы могут объединяться в различные оверлейные сети, которые, в свою очередь могут быть структурированными (structured) и неструктурированными (unstructured) [7].

1.3.1 Неструктурированные оверлеи

Неструктурированные оверлеи строятся в недетерминированном порядке, то есть в этих оверлеях не определена топология узлов. Если узел ищет данные на других узлах, то поисковый запрос распространяется по оверлею, стремясь охватить максимально количество узлов.

Неструктурированные оверлеи просты в построении. Если узел присоединяется к сети, то он соединяется с любым существующим узлом и использует его информацию для настройки соединения. Никакая статусная информация о присоединенном узле не передается в сеть, кроме как небольшой информации о маршрутизации от непосредственных соседей. Это делает неструктурированные оверлеи превосходным решением для случая большой текучести (churn) узлов – процесса присоединения/отсоединения узлов от сети.

Однако алгоритмы локации ресурсов (поиска) в сети становятся затратными. Если узел запрашивает ресурс, то запрос распространяется по сети по определенному алгоритму чтобы обойти максимально возможное (по некоторому критерию) количество узлов. Известны следующие алгоритмы поиска:

- поиска в ширину (BFS);

- алгоритм случайных блужданий (random walk) [37, 38, 44],
- поиск в глубину с итеративным углублением (iterative deeping) [19] и др.

Основу этих алгоритмов составляет распространение запросов всем соседним узлам. Узлы-соседи, принимающие запросы, возвращают результаты, а также последовательно пересылают запрос своим соседям. Этот процесс продолжается пока не выполнится условие окончания распространения запроса. Наиболее часто встречающимся условием является «время жизни» (Time-To-Live – TTL).

Для устранения стоимости распространения запроса может использоваться алгоритм случайных блужданий, когда узел передает запрос не всем соседям, а некоторым из них, выбранным в случайном порядке (рис. 1.5 [7]).

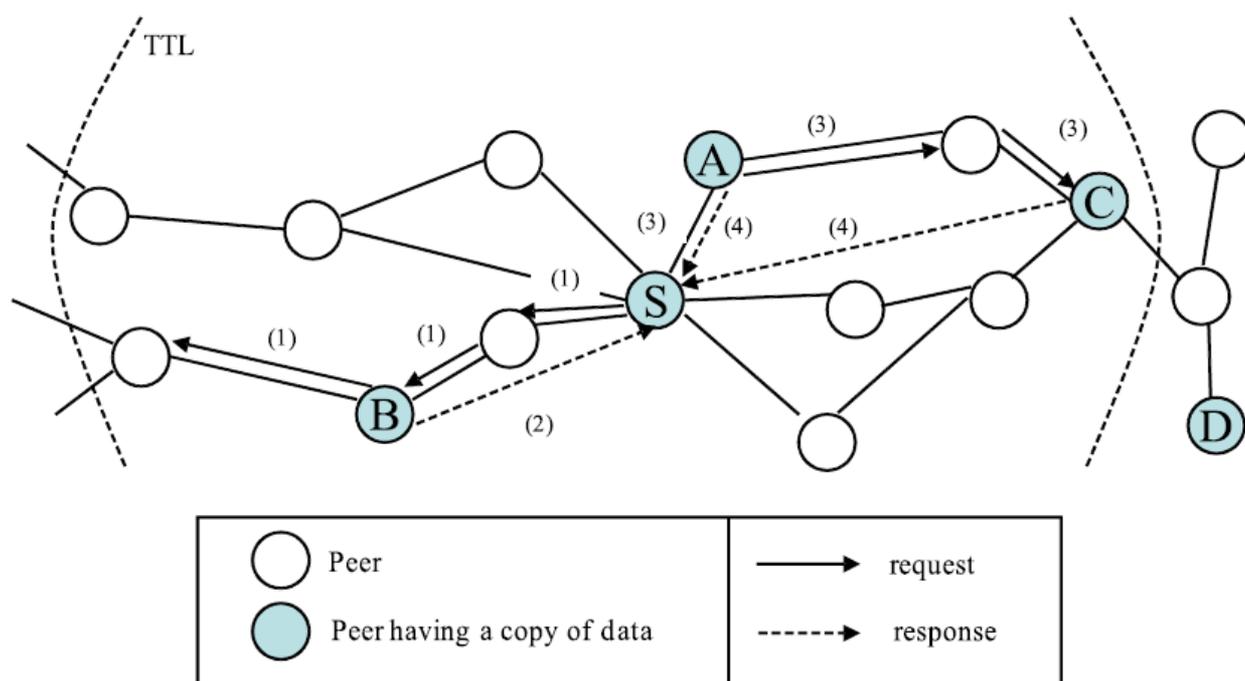


Рисунок 1.5 – Блуждающий поиск в неструктурированном оверлее

Узел S запрашивает сеть о трех копиях искомым данных. На первом шаге случайного выбора только узел B отвечает узлу S.

Однако алгоритмы поиска в ширину и случайных блужданий не обеспечивают полноценный, исчерпывающий поиск. Так узел D на рис. 1.5 не будет достигнут при TTL равном четырем.

1.3.2 Структурированные оверлеи

Структурированные оверлеи строятся таким образом, что узлы сети образуют конкретную топологию и в отличие от неструктурированных оверлеев узел не может произвольно менять свое местоположение, более того узел может обращаться только к неторному множеству соседей, определяемой топологией оверлея и идентификатором узла.

Появление структурированных оверлеев было обусловлено желанием ускорить поисковые запросы и повысить их эффективность. Для организации структурированных оверлеев используются следующие топологии [7]:

- кольцо;
- бабочка;
- гиперкуб;
- d-торж
- графы де Брюйна (De Bruijn Graphs);
- плакстоновские деревья (Plaxton Tree) и т.д.

Перечисленные топологии определяют некоторые пространства имен, в которых каждому узлу присваивается определённый уникальный идентификатор. Каждый узел поддерживает таблицу маршрутизации, связывающую идентификаторы и IP адреса соседних узлов в соответствии с топологией оверлея. Данная технология позволяет определять список узлов, за которые данный узел отвечает при поиске.

Программное обеспечение структурированных оверлеев должно реализовывать API, предоставляющее, как минимум, следующие примитивы:

- `join()` – функция присоединения вызывающего узла;
- `leave()` – функция удаления вызывающего узла из оверлея;
- `route(k, m)` – процедура маршрутизации сообщения `m` к корню узла `k`.

Как правило, протоколы структурированных оверлеев строятся на методах маршрутизации, основанной на ключах (key-based-routing – KBR). В свою очередь на KBR-протоколах функционирует ряд служб структурированных оверлеев, таких как:

- DHT, распределенные хэш-таблицы (Distributed Hash Tables);
- DOLR, децентрализованная служба определения объектов и маршрутизации (decentralized object location and routing);
- CAST, службы групповых коммуникаций, обеспечивающие присоединение и отсоединение узла к группе, а также рассылку широковещательных сообщений всем узлам-членам группы.

Подытоживая сказанное, можно сказать, что пиринговые системы строятся по многоуровневой архитектуре. Низший уровень, основанный на KBR, обеспечивает маршрутизацию сети. Вторым уровнем, обеспечивает специфические для P2P сетей службы (DHT, DOLR, CAST), и, наконец, третий уровень представляет из себя непосредственно P2P приложения.

1.4 Основные распределенные алгоритмы децентрализованных пиринговых систем

1.4.1 Основные принципы и проблемы разработки распределенных систем

Основные принципы разработки распределенных систем относятся к двум областям программного обеспечения информационных систем и распределенные алгоритмы. Ключевыми проблемами области программного обеспечения ИС являются: коммуникация, синхронизация и безопасность. Область распределенных алгоритмов вносит целый ряд проблем – отказоустойчивость, балансировка нагрузки, выбор лидирующего узла (leader election) и многое другое. Распределенный алгоритм разрабатывается для выполнения на узле с учетом координации и синхронизации с распределенными алгоритмами, запущенными на других узлах, для решения общих задач распределенной системы. Различают симметричные и асимметричные распределенные алгоритмы. Симметричные распределенные

алгоритмы выполняются на всех узлах распределенной системы, в то время как в ассиметричных алгоритмах на узлах могут выполняться разные компоненты.

К основным проблемам разработки распределенных алгоритмов следует отнести проблемы [7]:

- синхронизации;
- балансировки нагрузки;
- отказоустойчивости.

Фундаментальной проблемой распределенных систем является синхронизация доступа узлов к разделяемым ресурсам. Одним из основных механизмов, используемых для целей синхронизации, является использование взаимного исключения, при котором критическую часть может выполнять только один процесс распределенной системы. Большая часть распределенных алгоритмов использует именно взаимное исключение. Но, как известно из теории параллельного программирования, в этом случае могут возникать ситуации, называемая *deadlock* – взаимная блокировка. При взаимной блокировке один процесс, находясь в первой критической секции, ждет, когда второй процесс освободит вторую критическую секцию, в которой он и находится. Причем второй процесс также ожидает, когда первый процесс выйдет из своей критической секции. В ряде случаев для ликвидации взаимной блокировки приходится уничтожать один из заблокированных потоков в надежде ликвидировать *deadlock*. Однако такой принцип часто недопустим для распределенных систем. Распределенные алгоритмы в этом случае должны обеспечивать свое выполнение без возможности взаимной блокировки. Одним из возможных путей разработки алгоритмов без взаимной блокировки может быть реализация механизма «выбора лидирующего элемента» - *leader election*. По своей сути, *leader election* является другой общей для распределенных систем проблемой, требующей определения узла или группы узлов для решения специфических задач в распределенных системах.

Балансировка нагрузки относится к основным требованиям и требует равномерного распределения выполняемого кода и данных между узлами

распределенной системы. Код и данные должны переноситься с загруженного узла на узел с меньшей нагрузкой. У задачи балансировки нагрузки есть две основные метрики: время отклика (response time) и производительность (throughput). Время отклика определяет время между запросом к узлу и формированием узлом ответа. Под производительностью понимают количество задач, выполненных в определенное время. Основная цель балансировки нагрузки – уменьшить время отклика и увеличить производительность распределенной системы.

Целями обеспечения отказоустойчивости распределенной системы является обработка отказа вычислительного узла, пропадание связи между двумя узлами или отказ работы программного модуля на вычислительном узле. Одним из путей повышения отказоустойчивости является реплицирование кода и данных на другие узлы, чтобы в случае отказа реплики могли продолжить вычисление. Корректирующие узлы достигают соглашения, используя алгоритм консенсуса (consensus algorithm). Разработка алгоритма консенсуса является еще одной важной задачей в теории распределенных систем. Процедуры расстановки контрольных точек и восстановления по контрольным точкам периодически записывают на вторичных узлах состояние системы и, в случае сбоя, система может быть восстановлена в последнем сохраненном состоянии. Алгоритмы самостабилизации (Self-stabilizing algorithms) запускаются в случае возникновения сбоя и предназначены для достижения в течение нескольких шагов стабильного состояния системы.

1.4.2 Обобщенный список распределенных алгоритмов децентрализованных пиринговых систем

В результате анализа и систематизации рассмотренной выше информации был разработан обобщенный перечень алгоритмов, который должен реализовываться децентрализованной пиринговой системой. Не претендуя на полноту изложения материала, автором предложена следующая классификация указанных алгоритмов:

1) алгоритмы текучести (churn algorithms). Данная группа алгоритмов должна обеспечивать присоединение и отсоединение узлов к пиринговой сети.

Представителями этой группы являются следующие алгоритмы:

- функция присоединения вызывающего узла;
 - функция удаления вызывающего узла из оверлея;
 - функция перемещение узла по оверлею.
- 2) алгоритмы маршрутизации, среди них:
- процедура маршрутизации сообщения от одного узла к другому;
- 3) алгоритмы коммуникации точка-точка (передачи сообщений);
- 4) алгоритмы синхронизации, в том числе:
- leader election;
- 5) алгоритмы балансировки нагрузки, в том числе:
- функция подтверждения функционирования узла (поддержка heartbeat-сообщений);
- 6) алгоритмы отказоустойчивости, в том числе:
- создание точки восстановления узла;
 - реплицирование кода и данных на другие узлы;
 - алгоритмы консенсуса (consensus algorithm);
 - алгоритмы самостабилизации (Self-stabilizing);
- 7) алгоритмы доверия [7].

Приведенный перечень не является полным и в дальнейшем может быть расширен и конкретизирован в результате дальнейших исследований. Например, типовые процедуры узлов могут иметь специализированное значение на примере технологий распределенного реестра, blockchain технологий и криптовалюты.

Таким образом, пиринговые системы могут структурно делиться на три группы: централизованные, частично централизованные и децентрализованные. P2P сети, принадлежащие одной из структурных групп, имеют свои достоинства и недостатки, определяющие свойства безопасности и сложности распределенных алгоритмов. Логическая структура P2P сети определяется

оверлеями, которые могут быть структурированными и неструктурированными, что, в свою очередь, вносит дополнительные особенности и свойства пиринговых систем.

Пиринговым сетям, в общем, и децентрализованным, в частности, присуще реализация большого количества распределенных и нераспределенных (выполняемых узлом) алгоритмов, которые было предложено классифицировать по семи группам.

В соответствии с результатами изложенного выше материала в дальнейшем рассмотрим методы формального описания распределенных вычислений, специализированные языки программирования и P2P симуляторы. Данный материал позволит выявить основные элементы, которые могут быть применены при научных и инженерных исследованиях в области децентрализованных пиринговых систем.

2 МОДЕЛИРОВАНИЕ ДЕЦЕНТРАЛИЗОВАННЫХ ПИРИНГОВЫХ СИСТЕМ

2.1 Методы формального моделирования распределенных вычислений и систем

Пиринговые системы, как указывалось выше, должны быть многопоточными, распределёнными, мобильными [35]. В данном случае под мобильностью понимается два аспекта: мобильность кода (перемещение программного кода между узлами сети) и мобильность топологии (мобильные топологии, когда узлы сети могут менять свое местоположение). Поэтому для дальнейшего исследования необходимо определить круг методов формального описания и математического моделирования пиринговых систем. К методам моделирования вычислений, удовлетворяющим указанным требованиям, можно отнести [35]:

- λ -исчисление (λ -calculus), разработанное математиком Алонзо Чёрчем в 30-х годах прошлого века. λ -исчисление является формальной, полной по Тьюрингу системой, определяющей основу функциональных языков программирования. Данное исчисление может использоваться для формализации и анализа понятия вычислимости;
- π -исчисление (π -calculus) было разработано в 1992 г. Р. Милнером совместно с И. Парровом и Д. Уолкером как исчисление для моделирования и анализа параллельных вычислений в терминах взаимодействующих процессов, способных менять во времени топологию своего взаимодействия;
- исчисление (алгебра) процессов - группа методов формального моделирования параллельных систем. К указанным методам можно отнести исчисление взаимодействующих систем CCS (Calculus of Communicating Systems) [21], теорию взаимодействующих последовательных процессов CSP (Communicating Sequential Processes) [15], временные параллельные ограничения TCC (timed concurrent constraints) и др.

– модель акторов, предложенная в 1973 г. К. Хьюиттом и коллегами и впоследствии доработанная Грейфом, Клинкером, Ага, по сути, развивает λ -исчисление, определяет примитивы для создания и взаимодействия акторов и предоставляет соответствующую операционную семантику. Последовательные вычисления в акторе моделируются посредством λ -исчисления, а конфигурация актора моделирует взаимодействие и распределенное состояние;

– исчисление соединений процессов (join calculus) было предложено Ц. Фурнье и Дж. Гонтье в 1995 году как вид процессной алгебры, представляющую формальный базис для разработки распределенных языков программирования [17, 11];

– Кардели (Cardelli) и Гордан (Gordon) в конце 90-х годов прошлого века создали теорию исчисления окружения (Ambient calculus), в котором понятие мобильного окружения (mobile ambients) использовалось для моделирования кода, устройств и пользователей [5]. Авторы исчислений под мобильностью понимают как вычисления, выполненные на мобильных устройствах (сети с динамической топологией), так и мобильные вычисления (исполняемый код, способный перемещаться по сети). В ambient calculus определяется иерархическая структура окружений (ambients), содержащая процессы и другие окружения. Процессы ограничены в коммуникации только с аналогичными окружениями. Примитивы исчисления могут передаваться и приниматься от других окружений. Выражения исчисления окружений могут быть преобразованы в выражения π -исчисления и наоборот. Исчисление может быть применимо для моделирования взаимодействий в таких параллельных системах, как Интернет. Впоследствии исчисление окружений превратилось в семейство исчислений [2].

– Performance Evaluation Process Algebra (PEPA) – это стохастическая алгебра процессов, разработанная для моделирования компьютерных и коммуникационных систем Хилстоном (Hillston) [11]. Данное исчисление расширяет CCS алгебры Милнера и CSP, а PEPA модели сводятся к стохастическим процессам, а именно к непрерывному марковскому процессу.

Приведенный список не является исчерпывающе полным, поэтому существуют и другие методы формального моделирования распределённых систем и вычислений.

Наряду с рассмотренными методами при формальном описании пиринговых систем используются и классические для информатики методы, такие как графовые модели и теория множеств. Данная широкоизвестная группа методов также позволяет изучать и сравнивать модели централизованной и децентрализованной организации сетей [47].

Использование графовой модели [45] и теории множеств [48] позволяет изучать и сравнивать модели централизованной и децентрализованной организации сетей.

Сети Петри, являющиеся одной из разновидностей графовой модели, также активно используются при построении моделей распределенных систем и вычислений [35].

2.2 Языки программирования для распределенных и мобильных вычислений

На основе методов формального описания был разработан целый ряд языков программирования для построения распределенных и мобильных алгоритмов. Эти языки разработаны на основе формальных моделей, приведенных в предыдущем параграфе. Рассмотрим перечень языков, приведенных в [35].

2.2.1 Язык программирования Pi_{ct}

Программирование, основанное на абстракции процессов и каналов π -исчисления, может быть выполнено на языке программирования Pi_{ct}, предложенным Пирсом и Тернером (Pierce and Turner) в 2000 году [26]. Язык Pi_{ct} содержит типы, записи, соответствующие одному из варианту π -исчисления.

Расширение языка Pict, известное как Nomadic Pict, позволяет создавать распределенные и мобильные вычисления, также основанные на абстракциях π -исчисления. Nomadic Pict был предложен группой во главе с Сьюэллом (Sewell) в 2010 году [32]. Nomadic Pict расширяет Pict введением понятия мобильного агента, поддерживающего низкоуровневую (location dependent) и высокоуровневую (location-independent) коммуникацию.

Как и большинство высокоуровневых языков программирования Pict поддерживает специальный синтаксис для примитивных типов: booleans, characters, integers, and strings. Все они могут представляться процессами π -исчисления, аналогично их кодированию в λ -исчислении. Pict также предоставляет специальные примитивные каналы для встроенных операций (например +, *) и взаимодействия с окружением (например, каналы чтения/записи для представления стандартного ввода-вывода). Pict также поддерживает записи (records) и механизм регулярных выражений pattern matching.

2.2.2 Язык программирования SALSA

Программирование на акторах в настоящее время возможно на следующих высокоуровневых языках:

- SALSA [36];
- Erlang [3];
- Scala, предложен группой Мартина Одерски (Martin Odersky) в 2006 году.

В то время как языки Erlang и Scala основаны на функциональном программировании и фокусируются на программировании параллельных вычислений, язык SALSA, являясь объектно-ориентированным языком, фокусируется именно на создании распределенных и мобильных систем.

Язык SALSA напрямую используется для создания распределенных и мобильных вычислений. Среда выполнения SALSA состоит из службы имен

(name service) виртуальных машин Java, расширенных возможностями создания, миграции и взаимодействия сервисов.

2.2.3 Язык программирования JoCalm

JoCaml is a programming language implementing the join calculus, as well as the distributed join calculus, an extension with distribution and mobility primitives.

Программирование на основе join-паттернов как координированных абстракций может быть осуществлено на языке программирования JoCaml. Основными работами по данному языку являются работы Фессанта (Fessant), Конкона (Conchon), Мандела (Mandel), Марангета (Maranget) и др., выполненных с 1998 по 2008 год [20]. Паттерны объединений (join patterns) могут быть использованы как абстракции на других языках программирования высокого уровня. Например, язык Cω является расширением C# на основе join-объявлений (join definitions) [10]. По своей сути JoCaml является расширением языка Objective Caml, разработанного группой во главе с Лероем (Leroy) в 2008 [18], который в свою очередь является объектно-ориентированным диалектом функциональных языков семейства ML.

Аналогичные разработки проводились в СССР и России. Так в Институте программных систем РАН ведутся работы над языками T++, T# и MC# [43].

Язык T++ [24] представляет собой расширение языка C++, разработанное учеными России и Белоруссии в рамках суперкомпьютерной программы СКИФ. Программы, написанные на языке T++, выполняются так называемой T-системой – системой параллельного программирования с открытой архитектурой, которая поддерживает автоматическое динамическое распараллеливание программ [24]. При программировании на T++ программист не использует явные параллельные конструкции. Участки программы, выполняемые параллельно, обозначаются с использованием специальных указаний – расширений T++, представляющих ряд ключевых слов.

Языки T# и MC# [53] являются расширениями языка C#. Язык программирования MC# предназначен для написания параллельных и

распределенных программ. MS# использует модель асинхронного параллельного программирования, введенной в языке Polyphonic C#. В этой модели предложены высокоуровневые параллельные конструкции, которые превращают объектно-ориентированный язык C# в язык параллельного программирования.

Также имеется достаточно большой список языков программирования, поддерживающих разработку параллельных программ через свои библиотеки. К наиболее популярным языкам данного типа можно отнести Java и C++.

Таким образом были рассмотрены формальные модели и языки программирования, позволяющие проектировать и строить распределенные системы и алгоритмы, применяемые при построении децентрализованных пиринговых систем. Однако для проверки адекватности распределенных моделей и алгоритмов нужны средства компьютерного имитационного моделирования, рассчитанные на анализ пиринговых систем, насчитывающих миллионы узлов.

2.3 Симуляторы для исследования пиринговых систем

В предыдущей главе были рассмотрены основные виды пиринговых систем и определены основные распределённые алгоритмы, применяемые в децентрализованных P2P системах.

Приведем основные критерии, которые предъявляются к P2P системам следующие критерии [28]:

- самоорганизация: сеть самостоятельно подстраивается на появление или исчезновение узлов из сети, без потери целостности всей сети. Узлы используют локальную информацию от своих соседей для организации сети;
- распределенность: нет центрального узла, управляющего поведением сети;

– масштабируемость: проблемы пропускной способности узла, перекрытия узлов, отказа узла устраняются благодаря самой организации сети, позволяя сети масштабироваться на миллионы узлов.

Необходимо отметить, что в настоящее время имеется ряд симуляторов, разработанных в основном в США, и позволяющих проводить компьютерное моделирование децентрализованных пиринговых систем. К таким симуляторам можно отнести программные продукты OverSim [33], PeerSim [25], Tribler [34] и др. Анализ возможностей приведенных симуляторов является предметом отдельного исследования. Мы остановимся на рассмотрении основных возможностей наиболее часто упоминаемых в научной литературе симуляторов P2P сетей.

2.3.1 Симулятор PeerSim

P2P симулятор PeerSim [25] основан на обработке событий (event-based). Симулятор разработан в рамках проекта BISON и выпускается под лицензией GPL с открытым исходным кодом. PeerSim написан на Java и предназначен исключительно для epidemic-протоколов. Симулятор поддерживает два вида моделирования: на основе циклов (cycle-driven) и на основе дискретных событий (discrete-event) для чего применяются два соответствующих механизма симуляции (simulation engines). Симулятор разрабатывался как масштабируемый и динамический симулятор для моделирования больших P2P сетей. С помощью PeerSim можно моделировать структурированные и неструктурированные оверлеи. С помощью PeerSim могут симулироваться пиринговые сети, состоящие из миллионов узлов. Симуляция на основе циклов может поддерживать сети до 10^6 узлов. Идентификаторы узлов представляют из себя целые числа.

Симулятор может настраиваться через определение пользовательских сущностей. PeerSim позволяет пользовательским сущностям заменять большинство predefined. Также симулятор имеет расширяемые и подключаемые компоненты. Для гибкой настройки симулятора используется

ASCII файл, содержащий пары «ключ-значение». Входная информация представляется в текстовом файле, поэтому пользователь не может ее изменять во время выполнения.

Симулятор не поддерживает распределенного моделирования. Он не имеет графического пользовательского интерфейса. PerrSim написан на языке программирования Java.

2.3.2 Симулятор GPS

GPS (General Purpose Simulator) [12] представляет собой платформу P2P моделирования. Для точного и эффективного моделирования P2P протоколов симулятор поддерживает управляемую событиями на уровне сообщений (message-level event driven) архитектуру. Для обеспечения переносимости (portability), простоты разработки и расширяемости симулятор GPS использует Java как язык разработки и моделирования.

GPS – симулятор, основанный на механизме дискретных событий. Развитие P2P сети не представляется как синхронная обработка событий в каждом такте синхронизации. Для этого применяется механизм возникновения событий и передачи сообщений. Точность симуляции поддерживается с учетом основных свойств сети и протокола (в данном случае TCP) без накладных расходов на моделирование уровня пакетов; это достигается за счет использования макроскопических моделей оценки производительности. GPS имеет поддержку средств генерации топологии пиринговых сетей.

В симуляторе GPS возможно моделирование любого P2P протокола. Симулятор имеет GUI интерфейс и поддерживает визуализацию сети.

2.3.3 Симулятор NeuroGrid

NeuroGrid – это P2P симулятор на основе Java, который был расширен для поддержки поискового моделирования для протоколов FreeNet, Gnutella и NeuroGrid. NeuroGrid представляет собой однопоточный дискретный симулятор событий. Он имеет файлы свойств, которые позволяют пользователю изменять

параметры моделирования. Пользователь может указать тип протокола для имитации, количество поисковых запросов для имитации и тип предпочтительного пользовательского интерфейса (например, графический интерфейс на основе апплетов). Апплет отображает поисковые сообщения, отправляемые поисковым узлам на каждом шаге. Статистика (например, количество проанализированных сообщений и состояния моделирования) может быть сохранена в файлах для последующего анализа. NeuroGrid Simulator предполагает, что расстояние между узлами является постоянным - сообщения с одним и тем же TTL отправляются по сети параллельно. После того, как поисковое сообщение отправляется другим определенным узлам, узлы, получившие сообщения, поочередно пересылают сообщения из-за однопоточной конструкции симулятора. Кроме того, до тех пор, пока событие поиска не завершится, будет активным другое новое событие поиска (последовательное выполнение событий поиска). NeuroGrid позволяет пользователю указывать количество узлов для имитации, начальное количество соединений для каждого узла, количество поисков, которые необходимо выполнить и начальную топологию сети (только кольцевые или неструктурированные оверлеи). Количество узлов в сети при типовых архитектурах персональных компьютеров не многим превышает 300 000 узлов.

2.3.4 Симулятор P2PSim

Основными задачами симулятора P2PSim являются:

- упростить понимание исходного кода протоколов одноранговых сетей;
- упростить сравнение различных протоколов;
- обеспечить разумную производительность.

P2PSim - один из немногих одноранговых симуляторов, использующих многопоточность в симуляции. P2PSim использует очередь событий для хранения ожидающих событий, отсортированных по метке времени.

P2PSim - это дискретный симулятор событий, который используется только для моделирования структурированных оверлеев. P2Psim может имитировать сбои узлов. Симулятор поддерживает итеративный и рекурсивный поиск. Для идентификации узлов используется 160-битное хеширование SHA-1. В симуляторе нет поддержки распределенного моделирования. P2PSim не позволяет анализировать большие колебания полосы пропускания каналов.

Список симуляторов P2P сетей может быть продолжен. Так автором были найдены материалы, касающиеся следующих симуляторов: RealPeer, 3LS, Query Cycle, Serapis, FreeNet, FreePastry.

Даже поверхностный анализ изложенного материала, касающегося симуляторов пиринговых сетей, позволяет сделать вывод, что этот вопрос не является хорошо проработанным. Так явно видно, что необходима разработка симуляторов, способных моделировать пиринговые сети как с различными видами структурированных оверлеев, так и сети, построенные на основе неструктурированных оверлеев. Симулятор должен давать возможность моделировать каналы связи с большой девиацией полосы пропускания. Также можно отметить необходимость простого и динамичного моделирования поведения узлов сети. И, наконец, большой задачей является разработка симуляторов, которые построены на распределённых архитектурах и могут моделировать сети из десятков, сотен и более миллионов узлов.

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы был выполнен анализ научной литературы по теме моделирования децентрализованных пиринговых систем.

Были обобщены понятия и классификация понятий, касающихся теории распределённых вычислений и систем в части касающейся P2P сетей. Было показано место пиринговых сетей в классификации распределённых систем. Была рассмотрена классификация P2P сетей с точки зрения централизации архитектуры. Данная классификация включает три вида P2P сетей: централизованные, частично централизованные и децентрализованные сети.

В свою очередь независимо от архитектуры P2P сети могут образовывать различные логические топологии - оверлеи. Было показано, что различают два вида оверлеев: неструктурированные и структурированные. Структурированные оверлеи могут строиться по различным топологиям, таким как: кольцо, бабочка, гиперкуб, d-тор, графы де Брюйна и т.д.

Несмотря на различные архитектуры и топологии, пиринговые сети должны поддерживать различные группы распределённых алгоритмов, которые бы решали основные проблемы распределённых вычислений, а именно проблемы: синхронизации, балансировки нагрузки и отказоустойчивости. Для решения этих проблем были отмечены алгоритмы поиска консенсуса и выбора лидирующего звена. Типовые алгоритмы, используемые P2P сетями, было предложено объединить в семь групп, среди которых: алгоритмы текучести, маршрутизации, коммуникации, синхронизации, балансировки нагрузки, отказоустойчивости и алгоритмы доверия.

В первой части второй главы был рассмотрен список методов формального моделирования распределённых, в том числе пиринговых, систем. Как было показано, основу рассмотренных методов составляет λ -исчисление. Однако настоящий прорыв в области моделирования и анализа параллельных и распределённых вычислений осуществило π -исчисление, разработанное в 1992 году. Как показал анализ научной литературы, в дальнейшем на основе π -

исчисления были разработаны более полные и современные подходы, такие как исчисление акторов, исчисление соединений процессов и др.

На базе методов формального описания распределенных систем на протяжении двух десятилетий разрабатывались специальные языки программирования распределенных вычислений. В работе кратко рассмотрены такие языки как Pict, SALSA, JoCalm. Также было отмечено их отличие от таких популярных языков программирования, позволяющих писать распределённые алгоритмы, как Java и C++.

В завершении исследования была рассмотрена проблема компьютерного имитационного моделирования. Для этой цели был разработан достаточно большой ряд P2P симуляторов, ориентированных на решение различных топологий и задач в области P2P сетей. В работе были приведены основные сведения о возможности таких симуляторов как PeerSim, GPS, NeuroGrid, P2PSim. Также было отмечено, что существует явная проблема разработки современного симулятора, способного моделировать пиринговые сети, содержащие десятки и сотни миллионов узлов и поддерживающие основные свойства пиринговых сетей, таких как масштабируемость, мобильность и распределённость.

Выполненный обзор основан на анализе более чем 50 научных источников, среди которых 37 источников на иностранном языке, в том числе 29 статей и 5 монографий.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Abdallah A. E. Communicating Sequential Processes. The First 25 Years: Symposium on the Occasion of 25 Years of CSP, London, UK, July 7-8, 2004. Revised Invited Papers. – Springer Science & Business Media, 2005. – Т. 3525.
2. Ambient calculus // Электронный ресурс. 2018. - Режим доступа: <http://enacademic.com/dic.nsf/enwiki/10961853>
3. Armstrong J. Erlang—a Survey of the Language and its Industrial Applications //Proc. INAP. – 1996. – Т. 96. // Электронный ресурс. 2018. - Режим доступа: <https://pdfs.semanticscholar.org/8fce/0a789d4f8228c46734d26fda387468fb29f9.pdf>.
4. Baeten J. C. M. A brief history of process algebra //Theoretical Computer Science. – 2005. – Т. 335. – №. 2-3. – С. 131-146.
5. Cardelli L., Gordon A. D. Mobile ambients //International Conference on Foundations of Software Science and Computation Structure. – Springer, Berlin, Heidelberg, 1998. – С. 140-155.
6. Conchon S., Le Fessant F. Jocaml: Mobile agents for objective-caml //Agent systems and applications, 1999 and third international symposium on mobile agents. Proceedings. First international symposium on. – IEEE, 1999. – С. 22-29. // Электронный ресурс. 2018. - Режим доступа: https://www.researchgate.net/profile/Fabrice_Le_Fessant/publication/2636314_Jocaml_mobile_agents_for_Objective-Caml/links/553666be0cf20ea35f132381.pdf
7. Distributed systems : design and algorithms / edited by Serge Haddad ... [et al.] // John Wiley & Sons, Inc, 2011. – 324 p.
8. Ebrahim M., Khan S., Mohani S. S. U. H. Peer-to-peer network simulators: an analytical review //arXiv preprint arXiv:1405.0400. – 2014. // Электронный ресурс. 2018. - Режим доступа: <https://arxiv.org/ftp/arxiv/papers/1405/1405.0400.pdf>
9. Erciyes K. Distributed graph algorithms for computer networks. – Springer Science & Business Media, 2013.

10. Fournet C. et al. JoCaml: A language for concurrent distributed and mobile programming //International School on Advanced Functional Programming. – Springer, Berlin, Heidelberg, 2002. – С. 129-158. // Электронный ресурс. 2018. - Режим доступа: <https://pdfs.semanticscholar.org/1599/1cd205f8cacb091e3548a6311b4dd9cf84df.pdf>.
11. Fournet C., Gonthier G. The join calculus: A language for distributed mobile programming //Applied Semantics. – Springer, Berlin, Heidelberg, 2002. – С. 268-332.
12. GPS - General Purpose Simulator for P2P network. // Электронный ресурс. 2018. - Режим доступа: <http://www.cs.binghamton.edu/~wyang/gps>.
13. Han J. Distributed hybrid P2P networking systems //Peer-to-Peer Networking and Applications. – 2015. – Т. 8. – №. 4. – С. 555-556.
14. Hillston J. A compositional approach to performance modelling. – Cambridge University Press, 2005. – Т. 12. // Электронный ресурс. 2018. - Режим доступа: <http://www.dcs.ed.ac.uk/pera/book.pdf>
15. Hoare C. A. R. Communicating sequential processes //Communications of the ACM. – 1978. – Т. 21. – №. 8. – С. 666-677.
16. Huraj L., Šimon M., Host'ovecký M. P2P Grid Environment as a tool for Network Integrity Testing //Актуальные проблемы современной науки. – 2015. – С. 288-291.
17. INRIA, Join Calculus homepage // Электронный ресурс. 2018. - Режим доступа: <http://moscova.inria.fr/join/index.shtml>
18. Leroy X. et al. The objective caml system release 3.11 //Documentation and user's manual. INRIA. – 2008. // Электронный ресурс. 2018. - Режим доступа: <http://ftp.nluug.nl/os/Linux/distr/zenwalk/source/extra/d/ocaml/ocaml-3.12-refman.pdf>
19. Li X., Wu J. Searching techniques in peer-to-peer networks //Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks. – 2006. – С. 613-642.

20. Mandel L., Maranget L. Programming in JoCaml (tool demonstration) //European Symposium on Programming. – Springer, Berlin, Heidelberg, 2008. – С. 108-111. // Электронный ресурс. 2018. - Режим доступа: https://link.springer.com/content/pdf/10.1007/978-3-540-78739-6_8.pdf
21. Milner R. Communication and concurrency. – New York etc. : Prentice hall, 1989. – Т. 84.
22. Milner R. Lectures on a calculus for communicating systems //International Conference on Concurrency. – Springer, Berlin, Heidelberg, 1984. – С. 197-220.
23. Moon S. J., Yoon C. P. Information retrieval system using the keyword concept net of the P2P service-based in the mobile cloud environment //Peer-to-Peer Networking and Applications. – 2015. – Т. 8. – №. 4. – С. 596-609.
24. OpenTS. Описание языка T++ // Электронный ресурс. 2018. - Режим доступа: <http://www.opents.net/index.php/ru/lang-tp>
25. PeerSim: A Peer-to-Peer Simulator // Электронный ресурс. 2018. - Режим доступа: <http://peersim.sourceforge.net>
26. Pierce B. C., Turner D. N. Pict: a programming language based on the Pi-Calculus //Proof, language, and interaction. – 2000. – С. 455-494.
27. Rizzo F. et al. Beekup: A distributed and safe P2P storage framework for IoE applications //Innovations in Clouds, Internet and Networks (ICIN), 2017 20th Conference on. – IEEE, 2017. – С. 44-51.
28. Roussopoulos M. et al. 2 P2P or Not 2 P2P? //arXiv preprint cs/0311017. – 2003. // Электронный ресурс. 2018. - Режим доступа: <https://arxiv.org/pdf/cs/0311017.pdf>
29. San Ting N., Deters R. A generic peer-to-peer network simulator //Proceedings of the 2002-2003 Grad Symposium, CS Dept, University of Saskatchewan. – 2003. // Электронный ресурс. 2018. - Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.6717&rep=rep1&type=pdf>

30. Santoro N. Design and analysis of distributed algorithms. – John Wiley & Sons, 2006. – Т. 56.
31. Santoro N. Design and analysis of distributed algorithms. – John Wiley & Sons, 2006. – Т. 56.
32. Sewell P., Wojciechowski P. T., Unyapoth A. Nomadic pict: Programming languages, communication infrastructure overlays, and semantics for mobile computation //ACM Transactions on Programming Languages and Systems (TOPLAS). – 2010. – Т. 32. – №. 4. – С. 12. // Электронный ресурс. 2018. - Режим доступа: <http://www.cl.cam.ac.uk/~pes20/npict.pdf>
33. The OverSim home page // Электронный ресурс. 2018. - Режим доступа: <http://www.oversim.org>.
34. The Tribler home page // Электронный ресурс. 2018. - Режим доступа: <https://www.tribler.org/P2PSimulator>
35. Varela C. A., Agha G. Programming Distributed Computing Systems: A Foundational Approach. – MIT Press, 2013.
36. Varela C., Agha G. Programming dynamically reconfigurable open systems with SALSA //ACM SIGPLAN Notices. – 2001. – Т. 36. – №. 12. – С. 20-34.
37. Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D. Information retrieval techniques for peer-to-peer networks //Computing in Science & Engineering. – 2004. – Т. 6. – №. 4. – С. 20-26.
38. Zhaoqing Jia, Ruonan Rao, Minglu Li, and Jinyuan You. Random Walk Spread and Search in unstructured P2P, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 2003.
39. Антинескул А. В., Безукладников И. И., Кон Е. Л. Использование скрытых каналов для решения задачи поиска партнёров в P2P VPN-сети // Вестник Поволжского государственного технологического университета. Сер.: Радиотехнические и инфокоммуникационные системы. – 2015. – № 3 (27). – С. 27-37.

40. Васильев И. Ю., Гайдамака Ю. В. Имитационная модель буферизации видеопотока в одноранговой сети с учетом геолокации и активности пользователей //Труды Второй молодежной научной конференции «Задачи современной информатики». – С. 58.

41. Виткова Л.А. Исследование распределенной компьютерной системы адаптивного действия // Научные технологии в космических исследованиях Земли. 2015. Т.7. №5. С. 44-48.

42. Гриценко А. В. Обзор принципа работы пиринговых сетей и изучение возможности их применения для коммутации в распределенных вычислениях //Science Time. – 2017. – №. 2. – С. 114-120.

43. Гузеев В.Б. Расширение объектно-ориентированных языков программирования параллельными конструкциями для многопроцессорных и распределенных систем автореферат диссертации на соискание ученой степени кандидата технических наук / Российский государственный социальный университет. Москва, 2009 Режим доступа <https://dlib.rsl.ru/viewer/01003493833#?page=1>.

44. Звягин М. Ю., Шамин П. Ю., Прокошев В. Г. Повышение эффективности алгоритма случайных блужданий путем ветвления блуждания в промежуточных узлах //Телематика-2007Ж труды XIV Всероссийской научно-методической конференции–СПб. – 2007. – Т. 1. – С. 269-271.

45. Князьков В.С. Введение в теорию графов [Электронный ресурс] / В.С. Князьков, Т.В. Волченская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 76 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73674.html>

46. Комаров И.И. Проектирование архитектуры инструментального средства моделирования поведения сети типа P2P / И.И. Комаров, А.А. Шлыков, М.В. Назыров и др. //Альманах научных работ молодых ученых XLV научной и учебно-методической конференции Университета ИТМО. – 2016. – Том 5. – С. 274-276.

47. Кручинин С. В. Типы децентрализованных сетей и вариант построения децентрализованной сети полного стека протоколов //Известия Волгоградского государственного технического университета. – 2016. – №. 11 (190).
48. Математика. Дискретная математика [Электронный ресурс] : учебник / В.Ф. Золотухин [и др.]. — Электрон. текстовые данные. — Ростов-на-Дону: Институт водного транспорта имени Г.Я. Седова – филиал «Государственный морской университет имени адмирала Ф.Ф. Ушакова», 2016. — 129 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/57348.html>
49. Мировой сетевой трафик: настоящее и будущее // Электронный ресурс. 2018. - Режим доступа: <https://nag.ru/articles/article/31463/mirovoy-setevoy-trafik-nastoyaschee-i-budushee.html>
50. Молчанов А.Н., Белов Ю.С. Обзор технологий распределенных вычислительных систем // Электронный журнал: наука, техника и образование. 2015. № 3 (3). С. 43-52.
51. Москалец Н. В., Поповская Е. О., Тарасов К. А. Оптимизация суммарного взвешенного времени обслуживания в пиринговой сети //Научный альманах. – 2016. – №. 6-2. – С. 110-116.
52. Наглядный индекс развития сетевых технологий // Электронный ресурс. 2018. - Режим доступа: https://www.cisco.com/c/ru_ru/about/press/press-releases/2017/06-09b.html
53. Сайт проекта МС#. // Электронный ресурс. — 2018. — Режим доступа: <http://www.mcsharp.net>.
54. Тель, Ж. Введение в распределенные алгоритмы: монография / Жерар Тель. – М.: Изд-во МЦНМО, 2009.–616 с.