

ОБЗОР СОВРЕМЕННЫХ РАСПРЕДЕЛЕННЫХ СИСТЕМ**Раздобудов С.А., Мартышкин А.И.***ФГБОУ ВО «Пензенский государственный технологический университет», Пенза,
e-mail: alexey314@yandex.ru*

В настоящей статье приводится достаточно подробный обзор и классификация современных распределенных вычислительных систем. Тематика распределенных систем остается актуальной и на сегодняшний день. Задача разработки распределенных систем является одной из наиболее сложных задач, которые стоят в настоящее время перед информатикой и программной инженерией. Выделяются положительные и отрицательные стороны распределенных вычислительных систем. Определены классификационные признаки, по которым ранжируются современные распределенные системы. Представленный обзор фиксирует современные методы построения, использования и развертывания распределенных систем, что необходимо, в основном, для развертывания будущего исследования в данном направлении. В конечном итоге результаты исследования позволят решать вопросы разработки и развертывания распределенных систем. Исходя из рассмотренных в статье опубликованных к настоящему времени статей, книг и патентов, можно сделать вывод, что тематика развертывания распределенных вычислительных систем, в рамках даже локальной сети, рассматривается недостаточно глубоко и детально. Ввиду вышесказанного, можно сделать вывод, что возможна реализация различных моделей и алгоритмов для распределенных вычислительных систем, позволяющих в какой-либо мере ускорить развертывание распределенных систем.

Ключевые слова: распределенные вычислительные системы, проектирование, реализация, классификация, облачные технологии

OVERVIEW OF MODERN DISTRIBUTED SYSTEMS**Razdobudov S.A., Martyshkin A.I.***Penza State Technological University, Penza, e-mail: alexey314@yandex.ru*

This article provides a fairly detailed overview and classification of modern distributed computing systems. The topic of distributed systems remains relevant today. The task of developing distributed systems is one of the most difficult tasks currently facing computer science and software engineering. The positive and negative aspects of distributed computing systems are highlighted. The classification criteria for ranking modern distributed systems are determined. This review captures the current methods of building, using and deploying distributed systems, which is necessary mainly for the deployment of future research in this direction. In the end, the results of the study will help solve the issues of development and deployment of distributed systems. Based on the articles, books, and patents that have been published so far, we can conclude that the topic of deploying distributed computing systems, even within a local network, is not considered in sufficient depth and detail. In view of the above, we can conclude that it is possible to implement various models and algorithms for distributed computing systems that allow to speed up the deployment of distributed systems to some extent.

Keywords: distributed computing systems, design, implementation, classification, cloud technologies

Тематика распределенных систем (РС) остается актуальной и на сегодняшний день. Связано это в первую очередь с тем, что с помощью РС возможно произвести анализ свойств полученных данных, которые невозможно по ряду каких-либо причин обработать с помощью только одного вычислительного узла. Создание эффективных, способных к расширению вычислительных систем позволяет в значительной степени повысить скорость обработки данных.

Цель настоящей работы – провести обзор современных РС и выделить их достоинства и недостатки.

Материалы и методы исследования

Под «распределенными системами» на сегодняшний момент может пониматься значительное число определений. Наиболее полное и вместе с тем понятное определение РС предложил в своей книге Э. Таненбаум [1]: «Распределенная система – это набор независимых компьютеров, представ-

ляющийся их пользователям единой объединенной системой». Следует отметить, что в данном определении оговариваются несколько моментов: во-первых, что все вычислительные устройства данной системы автономны, а во-вторых, что пользователи полагают, что работают в единой системе. Еще одно определение РС дано в работе [2]: «Распределенными системами называются программно-аппаратные системы, в которых исполнение операций (действий, вычислений), необходимых для обеспечения целевой функциональности системы, распределено (физически или логически) между разными исполнителями».

Каждый компонент РС может иметь различные характеристики, отсюда следует, что скорость обработки информации отличается на разных узлах системы. При передаче сообщений могут иметь место различные задержки в связи с ограниченностью пропускной способности сети, сложной топологией сети или другими факторами,

которые необходимо учитывать при реализации системы.

На рис. 1 показан пример РС, в которой программные компоненты передают сообщения между собой через компьютерную сеть.



Рис. 1. Распределенная система

К плюсам РС можно отнести следующие: соотношение цены и производительности; высокая производительность; возможность разделение ресурсов; масштабируемость; надежность.

Основной проблемой РС является то, что в общем случае компонент РС имеет доступ только к локальной информации его текущего состояния и к информации о прошлых состояниях остальных узлов, полученной из сообщений, то каждому компоненту недостаточно информации для построения глобального состояния всей РС в целом.

Другой проблемой является то, что каждый компьютер имеет доступ только к своему счетчику времени, а у всей системы в целом понятие общего единого времени определить невозможно, поскольку, даже обмениваясь сообщениями, узлам будет доступна лишь устаревшая информация о времени других компонент системы. Основным следствием этого является то, что события в системе не полностью упорядочены. В отсутствии времени невозможно установить порядок двух любых событий, они могут происходить параллельно в рамках доступной компонентам информации.

Также в РС отсутствует детерминизм, что является следствием того, что в системе существуют случайные задержки передачи сообщений и независимость исполнения компонент системы.

Один из подходов к анализу РС основан на построении частичного порядка между событиями, происходящими в ней. В общем случае частичный порядок – транзитивное антирефлексивное антисимметричное отношение.

Введем частичный порядок предшествования на событиях, как транзитивное замыкание следующих правил:

– Посылка сообщений предшествует его приему;

– События, принадлежащие одному компоненту, естественным образом упорядочены.

Глобальным состоянием РС называют совокупность состояний процессов и каналов в какой-то момент времени. В данной работе не рассматриваются каналы, которые имеют какое-то состояние важное для анализа выполнения системы. Поэтому далее под глобальным состоянием РС будем понимать только совокупность состояний всех процессов в определенный момент времени.

Из того что построенный порядок является частичным, а не полным, и из самого его определения следует, что в РС могут встречаться такие события, которые не связаны данным порядком.

Неопределенность в работе с РС заключается в том, что ни один процесс, входящий в нее, не может определить в каком взаимном порядке в реальности происходили такие события. Более того при каждом новом запуске РС взаимный порядок таких событий может меняться. Возможны запуски РС, когда первым произойдет событие из одного процесса, но также возможен вариант, когда первым происходит событие второе. Данная проблема неопределенности исполнения РС подробно разобрана в работе [1]. Именно эта неопределенность и ведет к тому, что в РС мы не можем просто вычислить предикат для выполнения.

Неопределенность выполнения задач в РС ведет к тому, что при двух запусках система может проходить через различные глобальные состояния. Множество всех возможных глобальных состояний, через которые может проходить исполнение системы удобно представлять в виде решетки.

Скалярные часы Лампорта. Скалярными или логическими часами Лампорта называется целочисленная функция из множества событий во множество целых неотрицательных чисел. Предположим, что каждый узел РС имеет целочисленную переменную, изначально проинициализированную 0. Перед посылкой сообщения, компонент увеличивает ее на единицу. При посылке сообщения к нему отправляющий узел добавляет значение своей переменной, а при приеме сообщения узел присваивает своей переменной максимум из полученного значения и значения собственной переменной и увеличивает ее на единицу. Значением вышеупомянутой целочисленной функции на событии является значение

переменной, принадлежащей тому же узлу, что и событие. Стоит заметить, что логическое время события не уникально в системе, оно уникально только в рамках своего потока.

Оказывается, что если в РС ввести частичный порядок предшествования на событиях, то имеет место следующее утверждение: если a предшествует b , то логическое время часов Лампорта события a меньше логического времени события b (обратное утверждение не верно). Рассмотрим простой пример, в котором один узел передает сообщение второму.

На рис. 2 изображена пространственно-временная диаграмма одного из возможных исполнений РС данного примера. Прямыми горизонтальными линиями на рисунке отмечены процессы выполнения, в данном примере отображены два процесса. Отметки на линиях процессов обозначают события, которые в этом процессе происходили. Точками со стрелками отмечены события отправки и получения сообщений. Если стрелка выходит из точки, то это событие отправки сообщения, иначе событие приема сообщения.

Первое событие узла 2 предшествует второму событию узла 2, так как они естественно упорядочены в рамках узла, и их временные метки тоже упорядочены соответствующим образом, временная метка 3 меньше, чем временная метка 1. Также второе событие узла 1 предшествует второму событию узла 2, так как это события отправки и получения сообщения соответственно, также и их временные метки сохраняют порядок, событие получения сообщения содержит временную метку 3 которая больше чем временная метка 2 соответствующая событию отправки этого сообщения.

Векторные часы Лампорта. Логическими векторными часами называется функция из множества событий во множество векторов, состоящих из неотрицательных целых чисел.

Далее будем считать, что вектор X меньше вектора Y , если каждая компонента вектора X меньше или равна соответствующей компо-

ненте вектора Y и существует хотя бы одна компонента вектора X , которая строго меньше соответствующей компоненты вектора Y .

Каждый узел имеет целочисленный n -мерный вектор (n – количество узлов в РС), проинициализированный нулями.

Перед посылкой/принятием сообщения, поток увеличивает свою компоненту вектора.

При посылке сообщения к сообщению отправляющий поток добавляет свой вектор, а при приеме сообщения поток присваивает своей переменной покомпонентный максимум из полученного значения и значения собственной переменной.

Значением описанной функции на событии является значение переменной, принадлежащей тому же потоку, что и событие. Стоит отметить, что векторное время уникально для каждого события.

Оказывается, что если в РС ввести частичный порядок предшествования на событиях, то имеет место следующее утверждение:

Событие A предшествует событию B , тогда и только тогда, когда логическое время векторных часов события A меньше логического времени события B .

Важным свойством векторных часов в РС с введенным частичным порядком предшествования оказывается то, что при сравнении векторов времени двух событий достаточно сравнивать только компоненты процессов, которым эти события принадлежат.

В данной работе векторная метка времени, при разборе протокола исполнения, ставится в соответствие одному событию, таким образом, мы можем говорить о процессе, которому соответствует данная метка времени. Компонента вектора, которая соответствует процессу, в котором произошло рассматриваемое событие, соответствует числу событий, произошедших в этом процессе раньше. Остальные компоненты вектора времени означают число событий других процессов, о которых данному событию известно, то есть с которыми у него определен порядок предшествования.

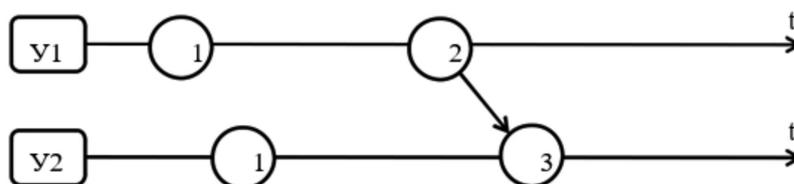


Рис. 2. Пространственно-временная диаграмма исполнения

При построении множества допустимых глобальных состояний метка времени ставится в соответствие одному узлу решетки возможных глобальных состояний. Тогда каждая компонента вектора времени означает количество событий, которое произошло в соответствующем процессе, чтобы наступило данное глобальное состояние.

Синхронные и асинхронные системы. Основной характеристикой способа взаимодействия подсистем РС является его синхронность или асинхронность. Стоит понимать, что параллельная работа подсистем не имеет никакого отношения к синхронности. Синхронное исполнение является более простым в реализации, поскольку, обрабатывая запрос вызываемой подсистеме гарантированно, что состояние вызывающей подсистемы не изменится до получения ответа. Пример синхронного вызова представлен на рис. 3.

В то же время для корректной реализации синхронного взаимодействия РС необходимо определять работает ли вызываемая подсистема или она аварийно завершена.

Асинхронные РС строятся аналогично синхронным, но вместо вызова и ожидания ответа выполняется отправка сообщения, а через некоторое время программа может проверить, прибыл ли ответ. Пример асинхронного вызова, с передачей сообщения вызываемому процессу без блокировки исполнения, представлен на рис. 4.

Реализация асинхронного взаимодействия позволяет избавиться от ограничений по времени на скорость выполнения операций, на задержку доставки сообщений и на скорость отклонения часов. Но множество проблем для этой модели не имеет решения.

Классифицировать РС сегодня можно по различным признакам [1]: по их размерам и способам администрирования (кластер, РС корпоративного уровня, глобальная система); по функциональности (вычислительные системы, информационные системы); по принципам их построения (РС объектов, распределенные файловые системы, РС документов).

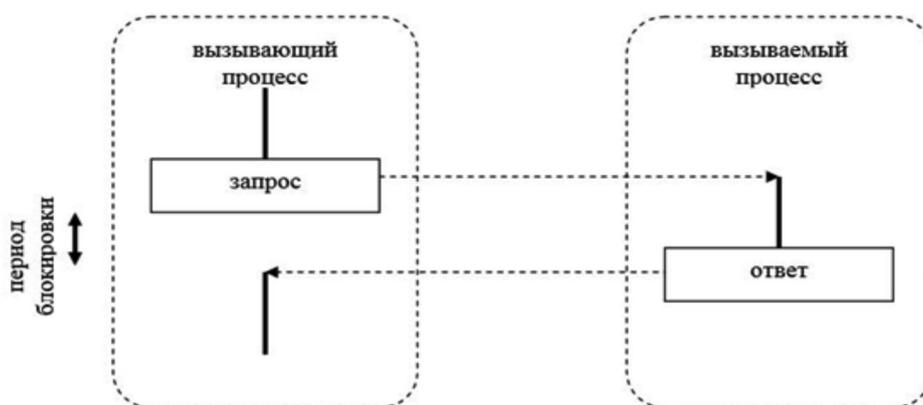


Рис. 3. Синхронное взаимодействие

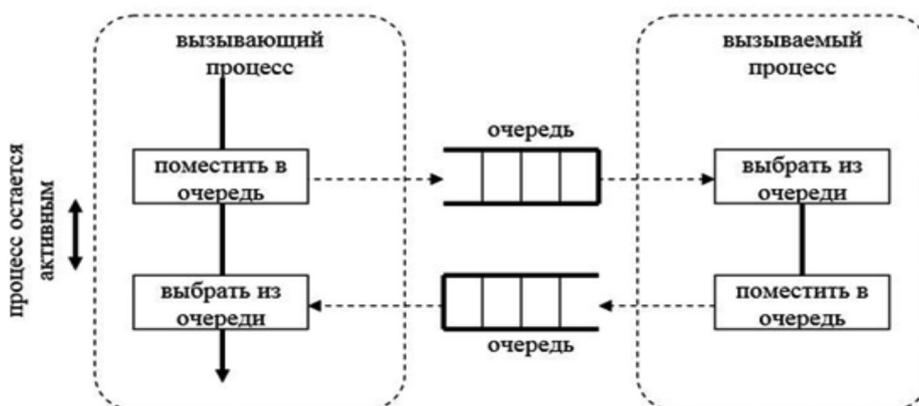


Рис. 4. Асинхронное взаимодействие

Настоящий обзор фиксирует современные методы построения, использования и развертывания РС. Такая фиксация необходима для развертывания будущего исследования в данном направлении. В конечном итоге результаты исследования позволяют решать вопросы разработки и развертывания РС.

В [2] подробно рассматриваются методики и инструментальные средства верификации РС. Фиксируются современное состояние методов для верификации РС и мощности их инструментальной поддержки. Большим плюсом данной статьи является широкий обзор инструментов для верификации РС и их подробный анализ.

Проблематика использования РС обзревается в [3]. Были выделены проблемы администрирования системы, проблемы балансирования нагрузки, проблемы восстановления данных в случае возникновения ошибок, проблемы ограниченности масштабируемости, проблема переносимости ПО.

В [4] излагаются конкретные подходы, которые пригодятся при реализации сервисов-производителей и сервисов-потребителей, участвующих в функционировании рабочих потоков на основе сообщений.

Связь в РС. Понятие «распределенная система» подразумевает, что компоненты такой системы распределены, т.е. удалены друг от друга. Очевидно, что функционирование подобных систем невозможно без эффективной связи между ее компонентами [5].

Взаимодействие в вычислительных сетях базируется на протоколах. Протокол – это набор правил и соглашений, описывающих процедуру взаимодействия между компонентами системы (в том числе и вычислительной) [6].

По тематике РС существует ряд патентов. Среди них можно выделить [7]. Данное изобретение обеспечивает формирование виртуальной сетевой топологии для разнообразных распределенных приложений, размещаемых на одной РС, не требуя физической перенастройки системы и конфигурирования проводных соединений внутри системы.

В настоящий момент перед множеством администраторов и связанных с ними спе-

циалистов по поддержке стоит сложная и зачастую продолжительная по времени задача по конфигурированию и развертыванию каждой вычислительной системы. С данной задачей в некоторой степени позволяет справиться [8]. В данном изобретении предоставляются варианты осуществления того, чтобы динамически конфигурировать, выделять и/или развертывать одну или более вычислительных систем на основе пользовательских требований.

Заключение

Исходя из рассмотренных ранее статей, книг и патентов, можно сделать вывод, что тематика развертывания РС, в рамках даже локальной сети рассматривается недостаточно глубоко. Ввиду вышесказанного, можно сделать вывод, что возможна реализация различных моделей и алгоритмов для РС, позволяющих в какой-либо мере ускорить развертывание РС. Именно данной теме и будет посвящено дальнейшее исследование. Решения, найденные в ходе дальнейших исследований, могут найти свое применение при разработке таких систем как «умный дом».

Список литературы

1. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
2. Бурдонов И.Б., Косачев А.С., Пономаренко В.Н., Шнитман В.З. Обзор подходов к верификации распределенных систем. Препринт 16. М.: ИСП РАН, 2006, С. 61.
3. Цветков В.Я., Алпатов А.Н. Проблемы распределенных систем // ПНИО. 2014. №6 (12).
4. Бернс Б. Распределенные системы. Паттерны проектирования. Питер., 2019. 224 с.
5. Радченко Г.И. Распределенные вычислительные системы. – Челябинск: Фотохудожник, 2012. 184 с.
6. Косяков М.С. Введение в распределенные вычисления. СПб: НИУИТМО, 2014. 155 с.
7. Хайдри Аамер (US), Седола Кент Д. (US). Генерация топологии виртуальной сети // Патент РФ № 2382398. Патентообладатель Майкрософт Корпорейшн (US). Заявка: 2004106718/09, 05.03.2004. Опубликовано: 20.02.2010, Бюл. № 5.
8. Линь Чи-Мин, Чжоу Шэн, Нандан Дургеш, Альбертсон Джеффри Ли. Динамическое конфигурирование, выделение и развертывание вычислительных систем // Патент РФ № 2429529С2. Патентообладатель Майкрософт Корпорейшн (US). Заявка: 2009107189/08, 30.08.2007. Опубликовано: 20.09. 2011, Бюл. №26.