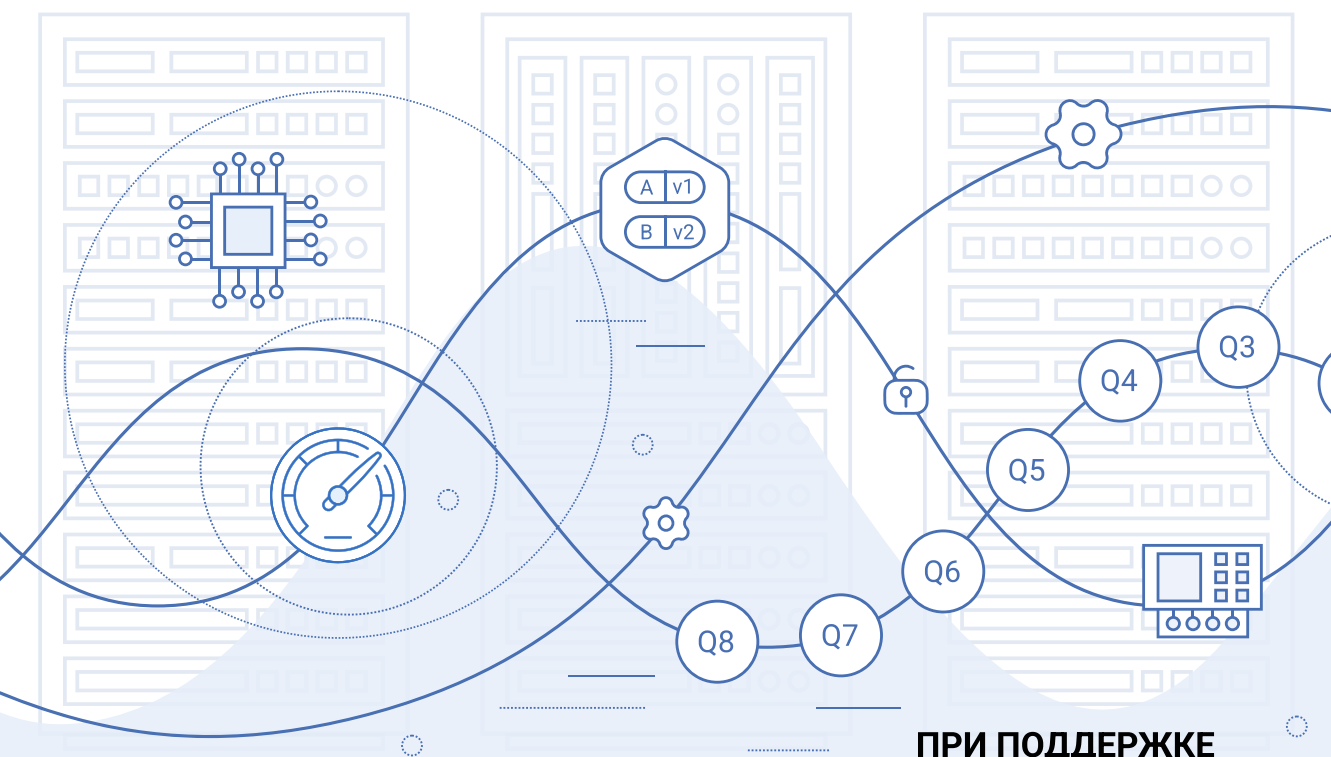


ВАДИМ ПОДОЛЬНЫЙ

АРХИТЕКТУРА ВЫСОКОНАГРУЖЕННЫХ СИСТЕМ

СИСТЕМЫ СБОРА ИНФОРМАЦИИ
РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ
СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ



ПРИ ПОДДЕРЖКЕ



В. П. Подольный

Архитектура высоконагруженных систем

Системы сбора информации

Распределенные системы управления

Системы реального времени

Москва

2022

УДК 004.6

ББК 16.35

П44 Подольный В.

Архитектура высоконагруженных систем. Системы сбора информации, распределенные системы управления, системы реального времени.

Издание первое. – М.: ООО «САМ Полиграфист» («OneBook»), 2022. – 160 стр, ил.

Книга дает представление о том, как проектируют высоконагруженные распределенные системы управления. Книга адресована всем тем, кто хочет разобраться, как устроены и создаются современные РСУ.

УДК 004.6

ББК 16.35

ISBN 978-5-00166-697-4

© Подольный Вадим

Оглавление

Благодарности	8
Об авторе	9
От автора: о чем эта книга	10
Обращение к читателю	12
Глава 1. Введение	13
Глава 2. Аппаратная архитектура	18
Глава 3. Программная архитектура	22
3.1. Данные и метаданные	22
3.2. Способы обмена данными	23
3.3. Транзакции	25
3.4. Распределенность	27
3.5. Ограничения распределенных систем	29
3.6. Единый формат данных	33
3.7. Операции с данными	33
3.8. Распределенные операции с данными	34
Глава 4. Принципы распределения данных	36
4.1. Распределенное хранение данных	36
4.2. Распределенная разделяемая память	39
4.2.1. Алгоритм с центральным узлом	41
4.2.2. Миграционный алгоритм	41
4.2.3. Алгоритм размножения	42
4.3. Избыточность данных	43
4.4. Сегментирование данных	44
4.5. Консолидация данных	47
Глава 5. Распределенная обработка данных	48
5.1. Узловая модель	51

5.1.1. Корневые узлы	51
5.1.2. Клиентские узлы	52
5.2. Способы обмена данными между узлами	53
5.3. Обработка сообщений	55
5.4. Обработка очередей	56
5.4.1. Приоритизированная обработка очередей	57
5.4.2. Многопоточная обработка очередей	58
5.5. Распределенная обработка очередей	60
Глава 6. Надежная распределенная архитектура	65
6.1. Резервирование: горячее, теплое и холодное	66
6.1.1. Резервирование корневых узлов	68
6.1.2. Режим мультиактивный	68
6.1.3. Режим мультиактивный с резервированием	69
6.2. Кэширование данных	70
6.3. Резервирование данных	72
6.4. Репликация данных	75
6.5. Синхронизация данных	77
6.6. Консистентность данных	82
6.6.1. Модели консистентности	83
6.6.2. Строгая консистентность	84
6.6.3. Слабая консистентность	84
6.6.4. Консистентность в конечном счете	85
6.6.5. Причинная консистентность	85
6.6.6. Последовательная консистентность	86
6.6.7. Консистентность по времени	86
6.6.8. Линеаризуемость	87
6.7. Современная классификация PCY	88
6.7.1. Типы PCY в зависимости от консистентности	90
6.7.2. Византийская задача	90

6.7.3. Блокчейн	91
6.8. Персистентность данных	92
Глава 7. Отказоустойчивость и высокая доступность	93
7.1. Диагностика и качество сервиса (QoS)	93
7.1.1. Метрики качества сервиса	94
7.1.2. Метрики приоритета	94
7.1.3. Метрики состояния	94
7.1.4. Композитные метрики	95
7.1.5. Метрика латентности heartbeat	96
7.2. Отказоустойчивость и надежность	97
7.2.1. Отказоустойчивая архитектура	100
7.2.2. Время восстановления	100
7.2.3. Катастрофоустойчивость	101
7.2.4. Архитектура высокой доступности	103
7.2.5. Доступность сервиса SLA	104
7.3. Дegradaция	105
7.3.1. Неустойчивое состояние	106
7.3.2. Элегантная и изящная деградация	107
7.3.3. Управляемая деградация	107
7.3.4. Дegradaция отклика	107
7.3.5. Дegradaция очередей	108
7.3.6. Дegradaция при информационном шторме	109
Глава 8. Балансировка нагрузки	110
8.1. Задачи балансировки	111
8.2. Типы балансировки	112
8.3. Логика балансировки	116
8.4. Балансировка без обратной связи	116
8.4.1. Циклическое распределение	116
8.4.2. Циклическое распределение взвешенное	117

8.4.3. Разновидности циклических алгоритмов	117
8.4.4. Минимум соединений	117
8.5. Балансировка с обратной связью	119
8.5.1. Циклическое распределение с ОС	120
8.5.2. Циклическое распределение взвешенное с ОС	121
8.6. Балансировка с приоритизацией	121
8.7. Теория массового обслуживания	122
Глава 9. Кластеризация	125
9.1. Принципы построения	125
9.1.1. Кластеры высокой доступности	126
9.1.2. Кластеры распределенной нагрузки	126
9.1.3. Вычислительные кластеры	127
9.1.4. Системы распределенных вычислений GRID	127
9.2. Кластерные вычисления	127
9.2.1. Параллельные вычисления	128
9.2.2. Катастрофоустойчивые кластеры	131
Глава 10. Сервисная архитектура	132
10.1. Масштабирование	132
10.1.1. Проблемы масштабирования	133
10.1.2. Линейное масштабирование	133
10.1.3. Функциональное масштабирование	134
10.2. Архитектура конечных сервисов	135
10.3. Жизненный цикл целевой системы	136
Глава 11. Варианты реализации	138
11.1. Распределенные СУБД	138
11.1.1. Ключ-значение	138
11.1.2. Объектная надстройка	139
11.1.3. Связи и отношения между объектами	141
11.1.4. Индексирование	142

11.1.5. Временные ряды	144
11.2. Критические информационные системы	145
11.2.1. АСУ непрерывного производства	147
11.2.2. АСУ дискретного производства	151
11.2.3. Цифровой двойник	152
11.2.4. Предиктивная аналитика	152
11.2.5. Граничные и туманные вычисления	153
11.2.6. Интернет вещей	154
Приложение 1. Принятые сокращения	155
Приложение 2. Список литературы	156
Приложение 3. Ссылки	158

Благодарности

Автор выражает искреннюю признательность коллегам за помощь в подготовке книги.

Прежде всего хотелось бы поблагодарить Клуб топ-менеджеров «Клуб директоров ИТ - 4 СИО.Ру», редакционную коллегию учебников по цифровой трансформации «4CIO» и «4CDTO», персонально Сергея Кирюшина, Феликса Карасева, Евгения Борисова, Владимира Определенова, Дмитрия Северова, Алексея Кравченко, Евгения Колесникова за полученный опыт разработки глав учебника 4CIO/4CDTO, мотивацию к созданию собственной книги и, конечно же, критику и отзывы.

Автор благодарит за активное участие в обсуждении книги, критику и ценные замечания — профильных специалистов Антона Жбанкова, эксперта в области создания аппаратной инфраструктуры высоконагруженных систем, и Сергея Шелудько, эксперта в области разработки распределенных высоконагруженных программных систем.

Автор выражает благодарность опытному старшему товарищу и автору книг по цифровизации АО «Концерн Росэнергоатом» «Цифровая трансформация», «Цифровой двойник» и «Центры обработки данных» Александру Прохорову за замечания, которые позволили повысить качество представленной книги.

Об авторе

Вадим Подольный

С 2000 года работает в области разработки программного обеспечения промышленных распределенных высоконагруженных систем управления. За 20 лет Вадим принял участие во многих крупных инфраструктурных проектах, связанных с технологической независимостью ключевых отраслей промышленности. Стал главным конструктором и руководил разработкой Российской программной платформы системы верхнего уровня АСУ ТП АЭС класса DCS/SCADA (ПО Распределенных Технологий Автоматизации Лицензированное, ПОРТАЛ) в ГК Росатом. С 2004 года на базе ПО ПОРТАЛ созданы и успешно эксплуатируются более 30 систем верхнего уровня АСУ ТП на 12 энергоблоках АЭС с реакторами ВВЭР-1000 (Калининская АЭС, Ростовская АЭС), ВВЭР-1200 (Нововоронежская АЭС, Ленинградская АЭС-2, Белорусская АЭС) и БН-800 (Белоярская АЭС). Вадим был главным конструктором и руководителем разработки специальной операционной системы «Заря», обеспечивающей задачи сопряжения автоматизированных систем управления Вооруженными силами РФ, которая была разработана Центральным научно-исследовательским институтом экономики, информатики и систем управления «ЦНИИ ЭИСУ», который, в свою очередь, выполнял функцию главного конструктора АСУ ВС РФ. Как опытный эксперт, Вадим участвовал в разработке и обеспечении кибербезопасности крупных распределенных автоматизированных систем ОАО «РЖД», ГК «Ростех», АО «ОРКК» и ряда коммерческих структур. Вадим является членом совета «Клуба Топ-менеджеров 4CIO», а также постоянным спикером многих отраслевых конференций, таких как HighLoad++, ИБ KBO, 4CIO и прочих.

Окончил факультет технической физики Национального исследовательского ядерного университета «МИФИ» (Московский инженерно-физический институт) по специальности «Ядерные реакторы и энергетические установки», а также аспирантуру Всероссийского НИИ по эксплуатации АЭС («ВНИИАЭС»). Получил дополнительное профессиональное образование в области АСУ ТП, разработки программного обеспечения и индустриальной кибербезопасности.

Связаться с Вадимом: vadim@podolny.ru (email), [@vpp01](https://t.me/vpp01)(tg)

От автора: о чем эта книга

Идея создания этой книги появилась после долгих обсуждений с техническими заказчиками подходов к разработке высоконагруженных распределенных систем. Всегда возникал один и тот же вопрос: как будем делать эту систему? Да вот так и будем. Заказчик уходил думать, пытаясь «сколхозить» решение собственными силами, а когда в очередной раз не получалось, возвращался и говорил: «Вот еще требования добавились, как делать-то будем?». Да все так же! И это могло повторяться снова и снова достаточно долго. Бывало так: приходишь, а там уже другой человек, и все начинается сначала. Мне это надоело, и я решил, что вместо объяснений буду вручать эту книгу. Конечно, это шутка, но в каждой шутке есть доля правды.

Эта книга не претендует на звание универсального свода знаний о высоконагруженной обработке данных или создании систем реального времени. Она лишь отражает некоторый опыт в этой области. Мой опыт в основном касается создания распределенных систем управления промышленными критическими информационными системами. В таких системах присутствуют сотни тысяч источников изменений данных и их потребителей. Сценарии управления зависят от характера и интенсивности этих изменений. Возможно, экспертам в области корпоративных ИТ-систем используемая терминология покажется несколько непривычной, но она появилась из-за сильно отличающихся и отчасти завышенных требований к промышленным системам.

Возникает вопрос: как обуздать этот хаос? Нужен *ordo ab chao*¹. А что, если из хаоса достаточно создать порядок, но не полный? Как вы обычно просите своего ребенка, чтобы привел в порядок свою комнату? Каковы метрики требуемого результата? С какого момента беспорядок можно считать в большей степени порядком, чем беспорядком? Так происходит и с данными в нагруженных системах. Чтобы с ними можно было работать, данные

¹ Порядок из хаоса

должны быть целостными. А какими должны быть метрики целостности (консистентности) данных, позволяющие однозначно сказать, готовы данные к обработке или нет? Ведь если управлять, например, опасным производством, основываясь на неконсистентных (устаревших) данных, может случиться авария. Моя книга именно об этом.

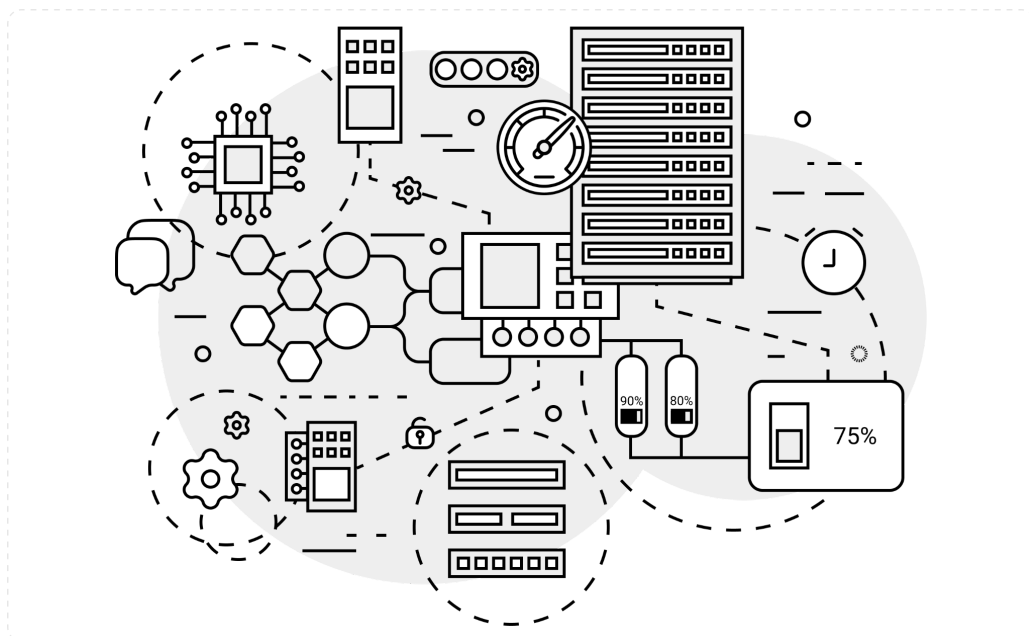
Книга дает представление о том, как проектируют высоконагруженные распределенные системы управления. Книга адресована всем тем, кто хочет разобраться, как устроены и создаются современные РС.

Обращение к читателю

Чаще всего в наше время понятие «высоконагруженные системы» используют применительно к компьютерной инфраструктуре массового предоставления услуг в Интернете 2.0. Причём массово используемые компьютерные системы — это и простейшие сим-карты размером с ноготь мизинца (да, сим-карты — это тоже компьютеры), и мощнейшие компьютерные установк-здания всем известных интернет-гигантов. Только энергопотребление отличается в миллиард раз. При этом кажется, что нагрузить до предела возможностей можно любую компьютерную систему. Тем сложнее составить собственное представление о высоконагруженных компьютерных системах и связанных с ними понятиях.

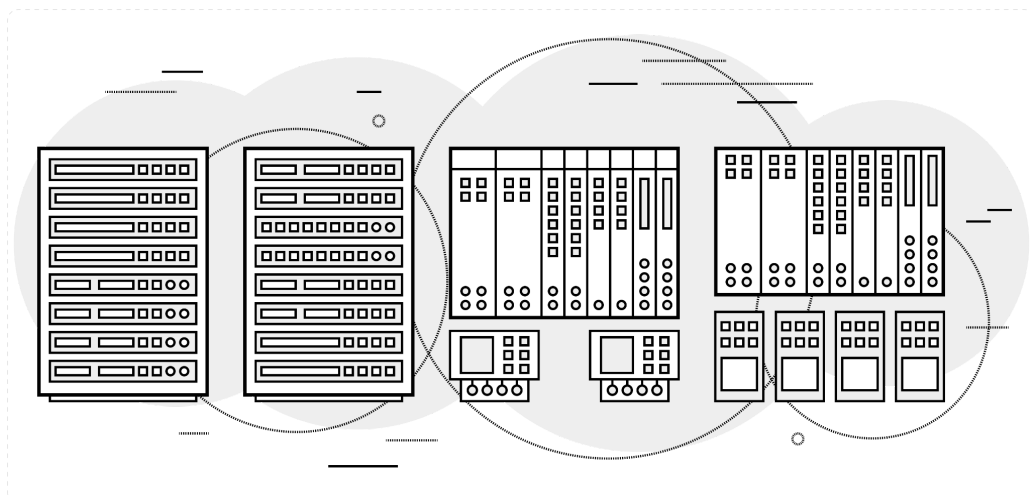
Автор помогает читателю охватить многочисленное множество понятий, связанных с высоконагруженными компьютерными системами, выступая как создатель и пользователь таких компьютерных систем, которые успешно управляют работой энергоблоков атомных электростанций, обслуживают и защищают инфраструктуру крупнейших предприятий страны. Специалисты оценят непривычную сжатость изложения тем, на которых автор предлагает сфокусировать внимание. Другие любознательные читатели в книге объемом всего лишь в 160 страниц обнаружат если не все, то большую часть сочетаний терминов, связанных с понятием высоконагруженных компьютерных систем. Надеюсь, что книга не оставит читателя равнодушным и подтолкнет его к более глубокому знакомству с высоконагруженными компьютерными системами, где и когда это будет ему наиболее интересно.

*Дмитрий Станиславович Северов,
экс-заместитель Министра связи и массовых коммуникаций РФ,
член совета Клуба топ-менеджеров «Клуб директоров ИТ - 4 СИО.Ру»*



Глава 1. Введение

Что такое высоконагруженная система (ВНС)? Очевидно, ВНС — это система, которая обрабатывает высокие нагрузки. Как обрабатывает? На чем обрабатывает? Насколько эффективно обрабатывает? На все эти вопросы есть ответы. Любая ВНС должна обрабатывать данные эффективно. Эффективно, то есть задействуя все доступные ресурсы аппаратного обеспечения, такие как вычислительная мощность всех ядер процессоров, оперативная память, сетевые устройства. Не должно быть такого, что, к примеру, из-за низкой эффективности (производительности) сетевых карт процессоры оставались загруженными не полностью. Такая ситуация означает, что аппаратное обеспечение подобрано неэффективно с точки зрения решения конкретной задачи обработки данных. Задачи обработки данных бывают самыми разными, их можно классифицировать. Для каждого класса задач обработки существует своя эффективная (оптимальная) аппаратная конфигурация. С экономической точки зрения аппаратную конфигурацию следует подбирать таким образом,



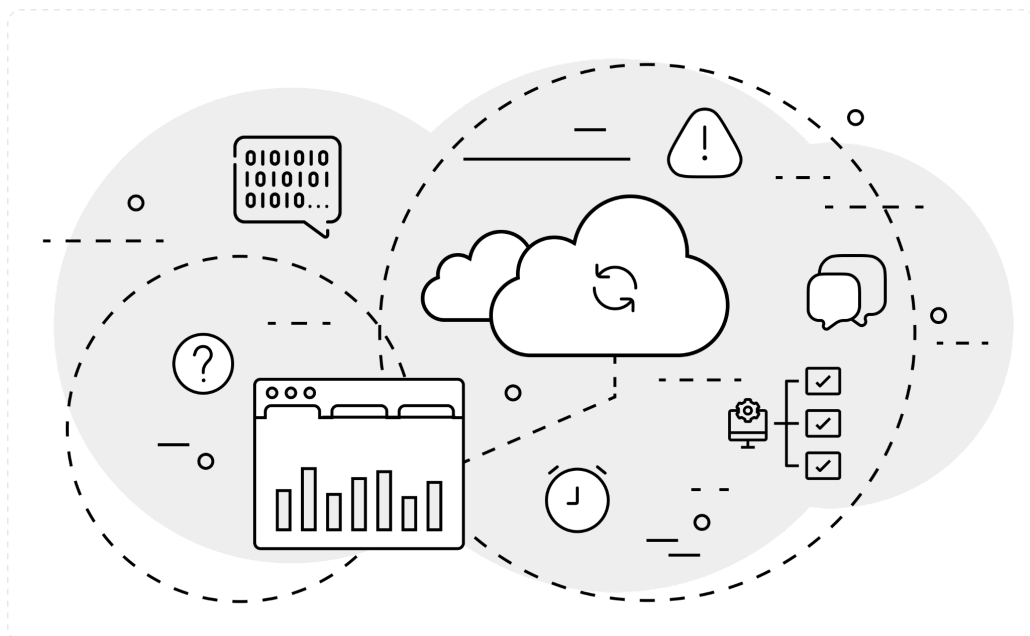
Глава 2. Аппаратная архитектура

В высоконагруженных системах, в том числе реального времени (ВНСПВ), важнейшей составляющей является аппаратная инфраструктура. В этой книге не будут подробно описаны, но будут выделены несколько моментов, которые в конечном итоге влияют на надежность, эффективность и стоимость инфраструктуры целевой системы. При проектировании аппаратной инфраструктуры для ВНСПВ следует учитывать несколько факторов.

Выбор типа железа

При проектировании рассматриваемых систем всегда возникает вопрос выбора типа используемого железа: использовать меньше серверов, но крутых, мощных и дорогих с резервированием компонентов, или больше, но дешевых и слабых без резервирования компонентов. Конечно же, на этот вопрос можно ответить, рассмотрев задачу не только с точки зрения аппаратного обеспечения (АО), но также в совокупности с программным обеспечением (ПО), потому что именно оно влияет на итоговую эффективность работы железа.

При выборе между «круто и дорого» и «завалить кучей дешевого железа» есть один правильный алгоритм действий:

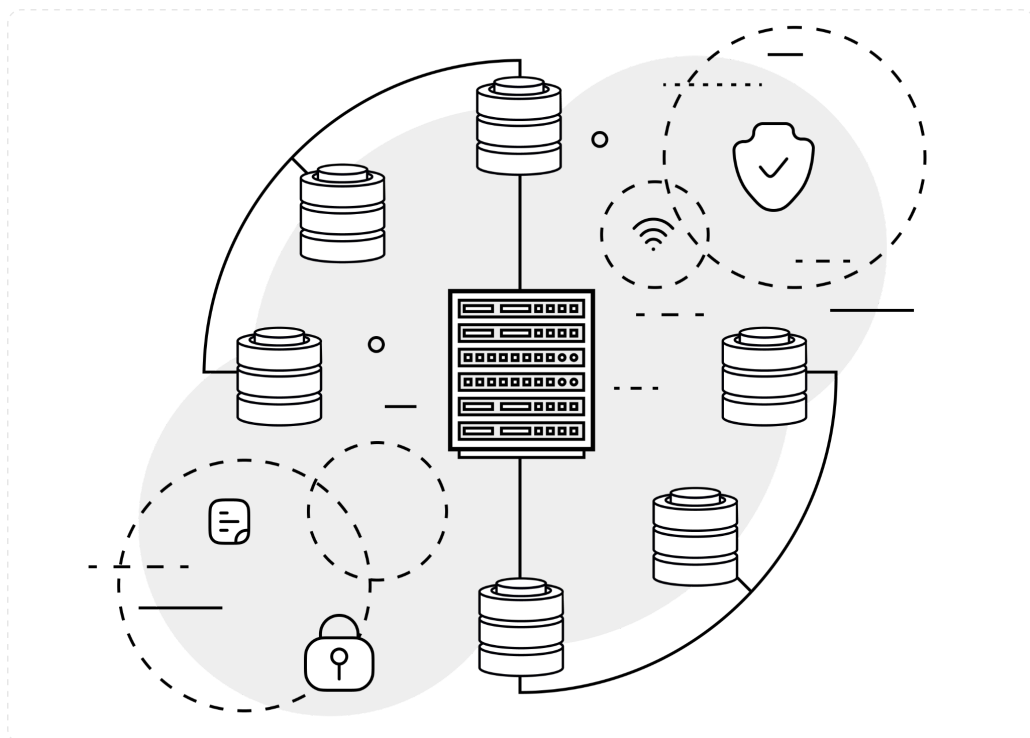


Глава 3. Программная архитектура

3.1. Данные и метаданные

Основная нагрузка в ВНС и СРВ связана с обработкой большого объема (потока) сообщений (данных, событий) из разнообразных внешних источников и внутренних составных компонентов системы. Принято выделять следующие типы событий:

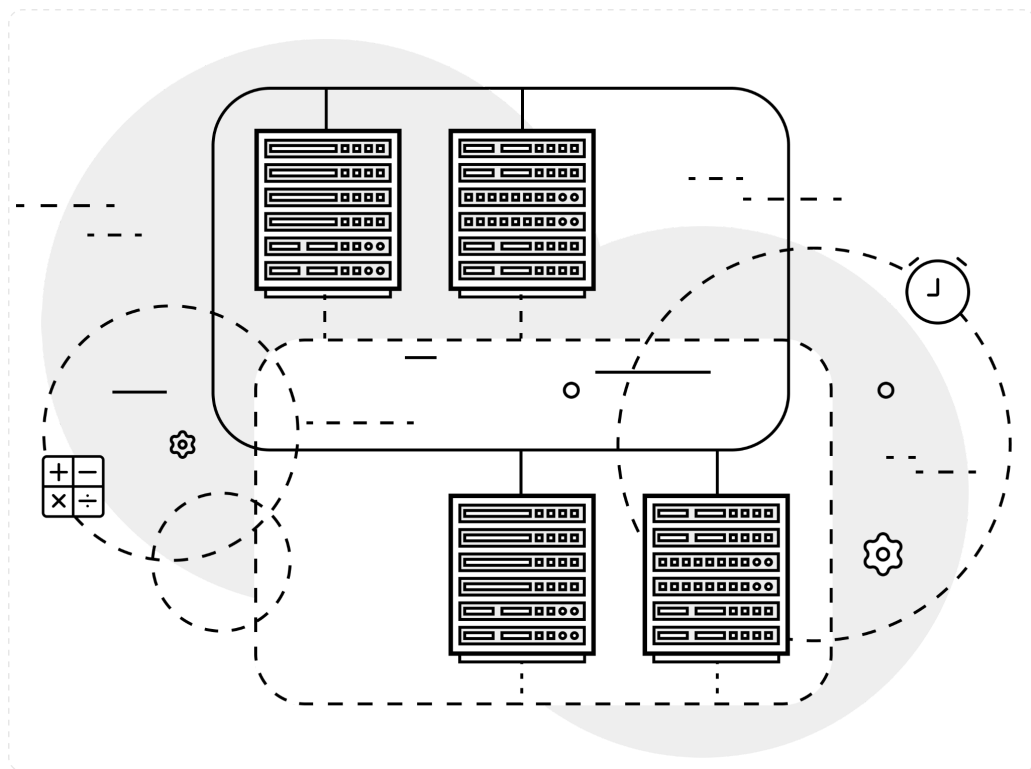
- асинхронные — полностью непредсказуемые события, такие как оповещения, тревоги, команды управления;
- синхронные — предсказуемые события, случающиеся с определенной частотой, такие как сообщения, связанные с обменом и синхронизацией данных;
- изохронные — регулярные события (разновидность асинхронных), случающиеся в определенный промежуток времени, такие как диагностические сообщения о состоянии компонентов системы (разд. [7.1.5](#)).



Глава 4. Принципы распределения данных

4.1. Распределенное хранение данных

В современных системах, сколь бы мощными ни были ресурсы узла (сервера), в какой-то момент они заканчиваются. В серьезных системах данные целесообразно обрабатывать на совокупности распределенных узлов. Это значит, что сначала данные необходимо распределить таким образом, чтобы их можно было удобно и эффективно обработать в распределенном режиме. Каждый такой узел обладает определенным объемом оперативной и постоянной памяти, определенными мощностями процессоров и пропускной способностью сетевых карт. В СРВ не принято задействовать постоянную память (ПЗУ), какими бы производителями ни были устройства. Распределение данных между



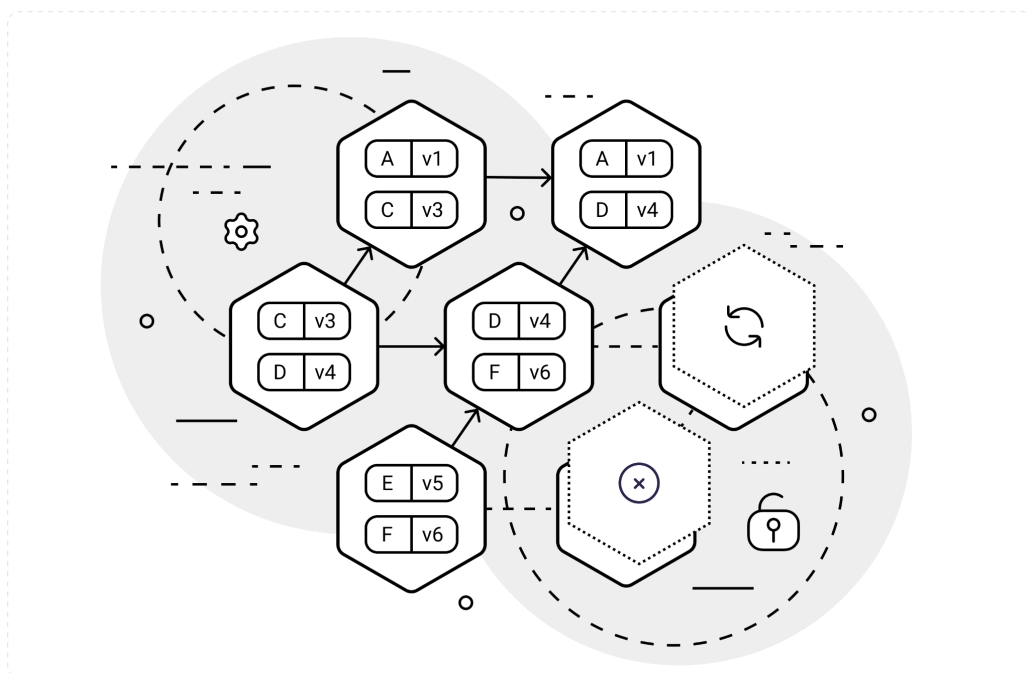
Глава 5. Распределенная обработка данных

Большинство современных ВНС представляют собой распределенные системы. Например, обширный класс распределенных ВНС, СРВ, можно отнести к распределенным системам управления (PCY²²). Современные вычислительные комплексы расчетного моделирования физических процессов построены как распределенные кластеры. Системы операторского сервисного обслуживания построены как распределенные системы обслуживания.

Такие системы отличаются друг от друга классом задач, возможностью смены конфигураций выполняемых задач «на лету», а также целевыми потребителями (пользователями).

Можно выделить следующие классы задач, решаемых PCY:

²² DCS, distributed control system, [#5-01](#)

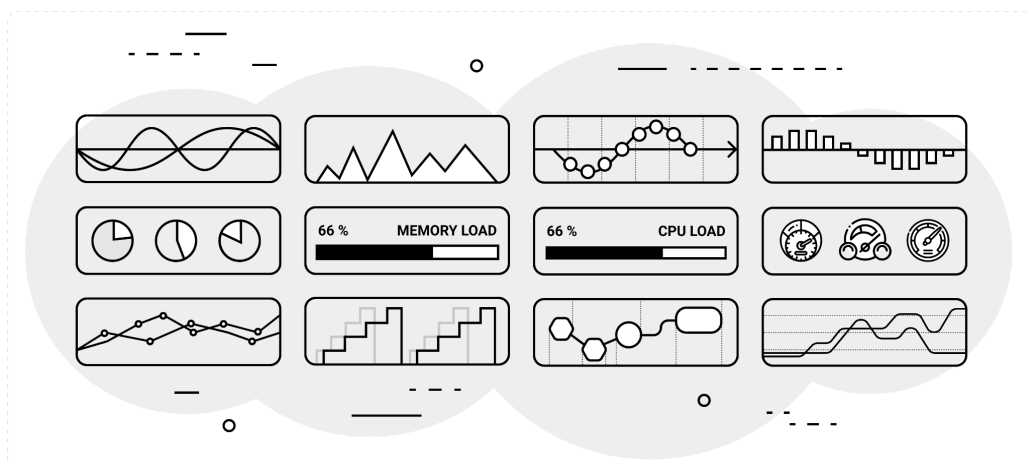


Глава 6. Надежная распределенная архитектура

Надежная архитектура подразумевает избыточность (разд. [4.3](#)) узлов и данных. Чтобы получить избыточность, необходимо выполнить ряд действий. Например, можно создать резервную копию на отключаемом ресурсе; вроде бы избыточность есть, но после отключения ресурса работать с резервной копией уже не представляется возможным.

Избыточность возникает, когда на пути от источника к потребителю данные проходят через промежуточный узел, где происходит кэширование (разд. [6.2](#)). Такие кэшированные данные вполне можно использовать как резервные.

Чтобы намеренно создать избыточность данных одного узла на другом, следует провести принудительную репликацию данных (разд. [6.4](#)). Репликация обычно представляет собой одноразовый процесс, если не предполагается создание полностью персистентной системы, в которой будут сохранены все измененные со



Глава 7. Отказоустойчивость и высокая доступность

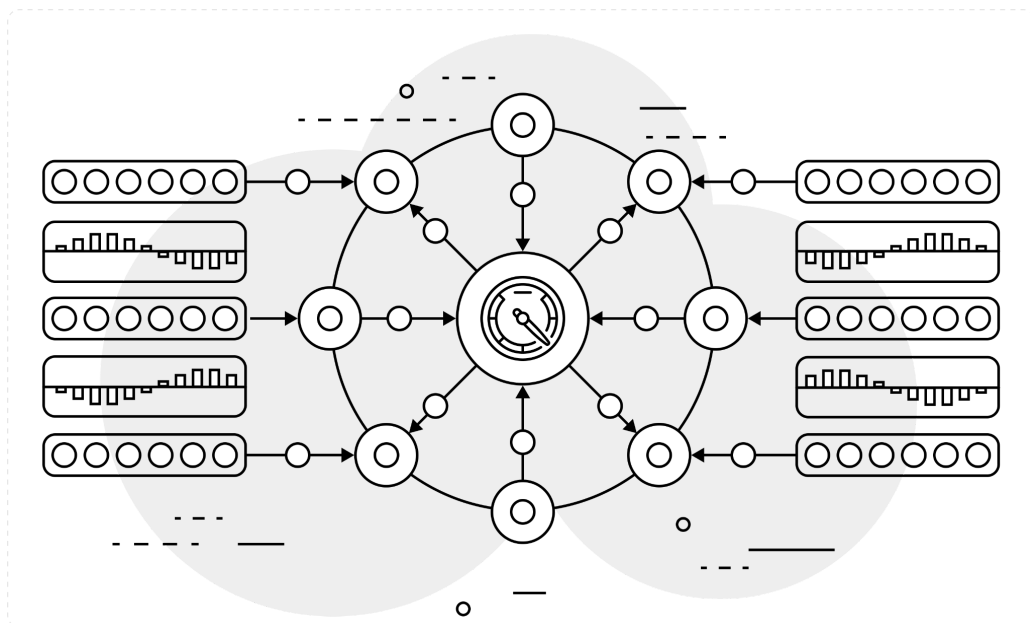
7.1. Диагностика и качество сервиса (QoS)

Обеспечение качества сервиса QoS⁴⁶ (обслуживания) является важнейшей задачей любой ВНС, в особенности СРВ. Существует множество разных определений QoS — например, «принцип предоставления различным классам трафика различных приоритетов в обслуживании» или «вероятность того, что сеть связи соответствует заданному соглашению о трафике (SLA), или обозначение вероятности прохождения пакета между двумя точками сети».

PCY обычно взаимодействуют не с трафиком низкого уровня, а с протоколами высокого уровня: начиная с протоколов транспортного уровня модели OSI⁴⁷ активно применяют протоколы сеансового уровня. В этом случае перечисленные выше определения можно свести к следующему: QoS — принцип предоставления различным классам сообщений различных приоритетов в обслуживании (разд. 5.3). Для присвоения сообщению определенного класса необходимо задать приоритет данных (переменных,

⁴⁶ Quality of Service, [#7-01](#)

⁴⁷ OSI, [#7-02](#)

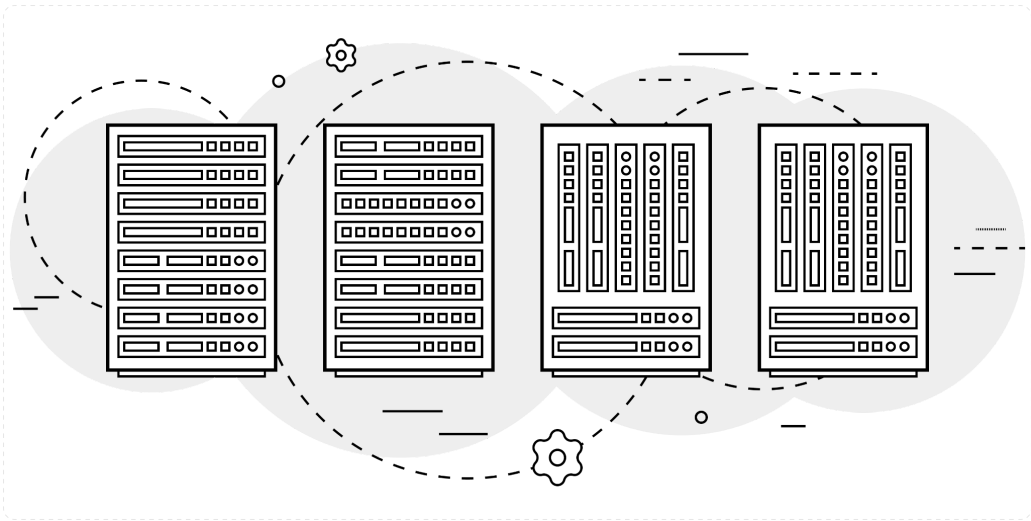


Глава 8. Балансировка нагрузки

Балансировка нагрузки⁶² (планирование нагрузки) РСУ является одной из сложнейших задач современных инфраструктур. Балансировка нагрузки — это метод распределения заданий между несколькими распределенными узлами с целью оптимизации использования ресурсов, сокращения времени отклика, горизонтального масштабирования (разд. [10.1](#)) узлов РСУ (динамическое добавление / удаление устройств), предотвращения перегрузки узлов, а также обеспечения отказоустойчивости (резервирования).

Архитектура схем и алгоритмов балансировки нагрузки весьма разнообразна, она зависит от целевой архитектуры системы. Например, архитектура балансировки нагрузки на узлы *mesh*-сети сложнее, чем архитектура балансировки нагрузки на узлы сети типа «звезда». В большинстве РСУ узлы, обеспечивающие основную нагрузку (на сервер), обслуживают сети типа

⁶² load balancing, [#8-01](#)



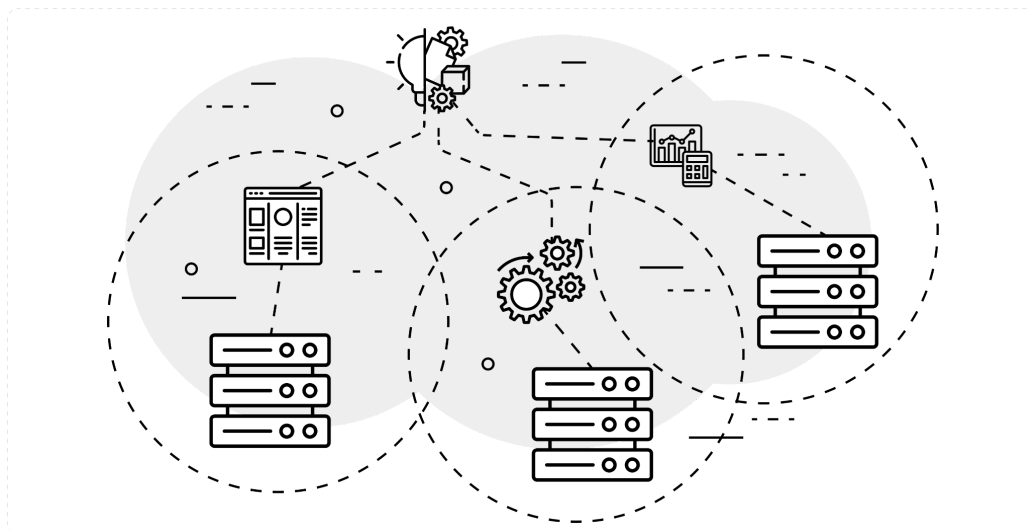
Глава 9. Кластеризация

В некоторых задачах PCY, CPB возможным и эффективным решением является объединение корневых узлов в единый вычислительный кластер. Кластерные вычисления необходимы для выполнения самых разнообразных задач, прежде всего для ускорения вычислений путем распределения задач по узлам.

9.1. Принципы построения

Вычислительным кластером называют объединение нескольких вычислительных узлов с помощью высокоскоростных каналов связи, с точки зрения пользователя представляющее собой единый аппаратный ресурс. Кластеризация подразумевает ряд жестких ограничений, применяемых к аппаратной составляющей задействованных узлов. Все узлы должны иметь идентичную аппаратную конфигурацию. Кластер всегда предполагает наличие выделенного интерконнекта, соединяющего кластерные (корневые) узлы.

PCY можно считать кластером, пока обеспечивается кворум кластера, то есть пока PCY содержит достаточное число активных узлов, позволяющих кластеру выполнить поставленную задачу.



Глава 10. Сервисная архитектура

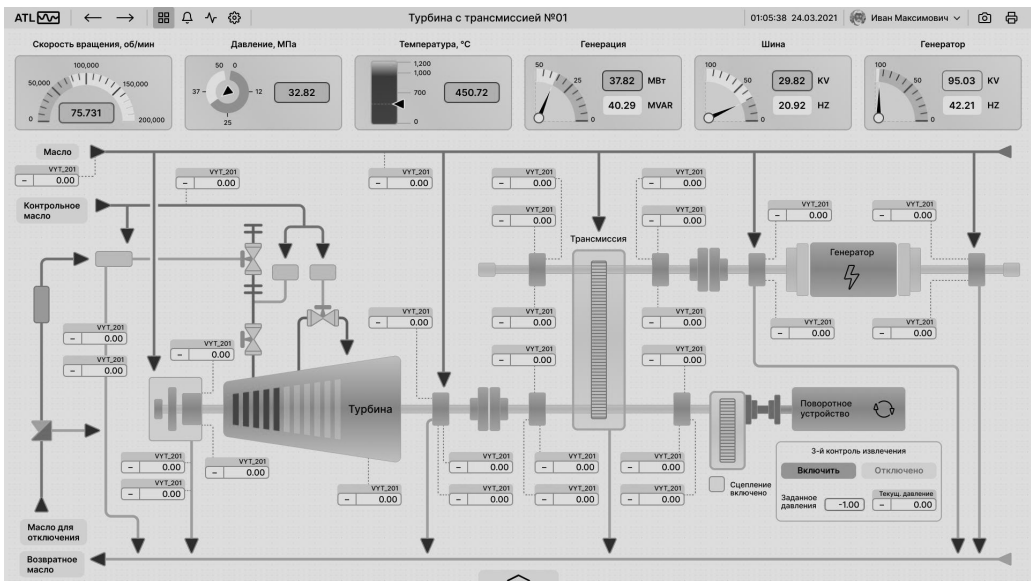
10.1. Масштабирование

Серьезная РСУ представляет собой большой и дорогостоящий инфраструктурный объект, который проектируют на десятилетия. Не исключено, что со временем система будет модернизирована, появятся новые дополнительные устройства, контроллеры и прочее оборудование. Это значит, что нагрузка на эксплуатируемую систему будет выше.

Необходимо заранее подготовиться к возможному масштабированию системы. Обычно масштабирование бывает необходимо в следующих случаях:

- увеличивается общий объем данных;
- увеличивается количество источников и разнообразие данных;
- усложняется логика обработки данных.

Во всех случаях масштабирование осуществляется путем добавления в систему новых узлов. Может случиться так, что



Глава 11. Варианты реализации

11.1. Распределенные СУБД

В этой книге не будут рассмотрены механизмы организации локального хранения данных на жестких дисках — будем считать, что архитектор найдет эффективный способ. Как уже было сказано выше, системы распределенного хранения и доступа к данным наиболее эффективно реализуют в архитектуре *in-memory*. Таким образом, при создании распределенной системы, на узлах которой будут задействованы жесткие диски, мы будем считать, что было обеспечено надлежащее кэширование данных в оперативную память для организации эффективной работы узла.

11.1.1. Ключ-значение

Сейчас самым простым⁸⁰ способом организации хранения распределенных данных является конструкция типа ключ-значение (*key-value*), которая представляет собой реализацию

⁸⁰ по мнению автора

Вадим Подольный

Архитектура высоконагруженных систем

Системы сбора информации

Распределенные системы управления

Системы реального времени

Литературный редактор – Надежда Тихомирова

Корректор – Максим Кузнецов

Иллюстратор – Анастасия Балякно

ISBN 978-5-00-166697-4



Подписано в печать 11.05.2022 г.

Формат 60x90/8. Бумага офсетная. Печать цифровая.

Заказ No 109257. Тираж 100 экз.

Отпечатано в типографии «OneBook.ru»

ООО «Сам Полиграфист»

109316, г. Москва, Волгоградский проспект, дом 42, корп. 5,
«Технополис Москва». www.onebook.ru