

## Синхронизация часов в распределенных многомашинных вычислительных системах. Часть II

© И.В. Ашарина

АО «НИИ «Субмикрон», Москва, Зеленоград, 124460, Россия

*Обоснована необходимость синхронизации часов в распределенных многомашинных вычислительных системах. Приведены основные определения, связанные с понятием синхронизации часов. Представлена классификация методов синхронизации часов. Рассмотрена одна из наиболее актуальных задач, которую необходимо решить при разработке сбое- и отказоустойчивых распределенных многомашинных вычислительных систем ответственного применения, особенно необслуживаемых, в частности космических, — увеличение сроков их активного существования. Вторая часть посвящена синхронизации в системах с византийскими неисправностями, что является сложной задачей, обусловленной особенностями модели неисправности. Процесс синхронизации в многокластерных и многокомплексных системах связан с многозадачностью таких систем, что делает процесс синхронизации еще более актуальным и многокритериальным. Рассмотрены современные технологии, обеспечивающие процесс синхронизации в системах ответственного применения.*

**Ключевые слова:** *распределенная многомашинная вычислительная система, сбое- и отказоустойчивость, синхронизация часов, динамическая избыточность, враждебная неисправность*

**Введение.** Синхронизация часов является одним из необходимых действий, связанных с разработкой современных сбое- и отказоустойчивых распределенных многомашинных вычислительных систем (РМВС) ответственного применения. Возрастающая сложность архитектуры современных РМВС приводит и к усложнению процесса синхронизации в них. Вместе с этим повышается актуальность вопросов синхронизации часов.

В первой части работы [1] приведены основные определения, связанные с процессом синхронизации, рассмотрены программные и аппаратные методы синхронизации, их особенности, преимущества и недостатки. Как показывает анализ литературных источников, для устранения внутренних ограничений аппаратного и программного подходов целесообразно использовать гибридную схему синхронизации, сочетающую приемлемые характеристики часов и минимальную избыточность аппаратного обеспечения.

В настоящей статье рассмотрены вопросы дрейфа часов и проблемы возникновения сбоя в момент синхронизации часов, способного нарушить процесс синхронизации, что с большой вероятностью приведет к нарушению работы системы в целом.

**Синхронизация в системах с византийскими неисправностями.** Синхронизация в сбое- и отказоустойчивых РМВС ответственно-го применения при наличии враждебных отказов — наиболее сложная и востребованная проблема синхронизации часов. Особенно эти вопросы актуальны и трудоемки при проектировании больших многопроцессорных систем реального времени, поскольку могут сопровождаться большими временными затратами и значительным количеством компонентов в системе.

С момента исследования, проведенного авторами [2], проблема синхронизации часов при наличии византийских неисправностей широко изучалась другими исследователями. Многим задачам требуется согласованное представление времени во всех узлах распределенной системы, и этого можно достичь только с помощью синхронизации часов.

Авторы [3] разработали алгоритм аппаратной синхронизации, который может синхронизировать процессоры в системе на двух разных уровнях:

1) уровень группы — все процессоры, работающие над одним и тем же заданием (взаимодействующими задачами), жестко синхронизированы друг с другом;

2) системный уровень — любые два процессора, принадлежащие к двум разным группам (и, следовательно, работающие над несвязанными задачами), слабо синхронизированы друг с другом.

Размер каждой группы зависит от следующих факторов:

- количества взаимодействующих задач, которые выполняются одновременно в этой группе;
- степени избыточности, необходимой для надежности, т. е. каждая из взаимодействующих задач будет выполняться более чем на одном процессоре в группе и результаты выполнения будут голосоваться.

Таким образом, группа процессоров состоит из избыточных кластеров, каждый из которых выполняет одну и ту же задачу.

Согласно [3], процессоры в кластере нуждаются в более жесткой синхронизации между собой, чем процессоры в группе. Это приводит к необходимости трех (вместо двух) типов синхронизации:

- наиболее жесткой внутрикластерной синхронизации;
- жесткой внутригрупповой синхронизации;
- слабой межгрупповой синхронизации.

При этом все процессоры в группе не должны быть так же жестко синхронизированы, как и в кластере. В соответствии с этим предположением термины *кластер* и *группа* употребляются в одинаковом значении [3].

Допуская такие предположения, следует учитывать, что подобные разночтения в терминах способны внести ошибочное понимание

процесса согласования. Это может привести к нарушению процесса согласования в системе в целом и, естественно, вызвать сомнение в ее сбое- и отказоустойчивости, т. е. в возможности применения таких методов при работе со сбое- и отказоустойчивыми РМВС ответственного применения.

Базовые аппаратные часы имеют, как правило, отклонение от реального времени, ограниченное некоторым фиксированным значением, но отклонение логических часов может превышать это значение. Следовательно, точность логических часов по отношению к реальному времени может оказаться ниже, чем аппаратных часов. Такое снижение точности бывает вызвано синхронизацией при наличии отказов. Скорость дрейфа неисправных аппаратных часов также может выходить за указанные границы, и тогда даже исправные логические часы могут быть вынуждены дрейфовать вместе с ними. Кроме того, неисправности способны изменить время доставки сообщений на неопределенную величину. Предложенные в [2, 4–9] алгоритмы синхронизации демонстрируют такое снижение точности часов.

В [10] показано, что процесс синхронизации не всегда может быть сопряжен с потерей точности. Авторы представляют алгоритм синхронизации, в котором логические часы имеют такую же точность, как и базовые физические часы. В работе показано, что никакой иной алгоритм синхронизации из перечисленных ранее не может обеспечить более высокую точность. В работах [2, 4–9] для каждой модели неисправности был представлен свой алгоритм, а авторы [10] предлагают унифицированное решение для синхронизации часов сразу для нескольких моделей неисправностей: сбоев, пропусков, так называемых византийских неисправностей, как с проверкой подлинности сообщений, так и без нее.

Для преодоления больших сложностей, связанных с разработкой оптимального алгоритма синхронизации в системе с произвольными отказами, авторы [10] сначала делают предположение, что система обеспечивает аутентификацию сообщений [6, 11], и в рамках таких гипотез строят достаточно простой алгоритм синхронизации. Затем подписанное сообщение заменяется ширококвещательным примитивом, который обеспечивает свойства аутентификации сообщения, которые требуются алгоритмом, предложенным в [12]. Это позволяет создать эквивалентный алгоритм для систем с произвольными неисправностями без цифровых подписей, который затем упрощается для случая сбоев и пропусков. В работе [10] приведен достаточно мощный математический аппарат, описывающий предложенные действия.

Несмотря на огромное количество публикаций, посвященных синхронизации часов, наиболее подходящим методом синхронизации в сбое- и отказоустойчивых РМВС ответственного применения пред-

ставляется описанный в [13], где взаимная синхронизация вычислителей посредством обмена сообщениями рассматривается как предваряющая взаимное информационное согласование в распределенной асинхронной вычислительной системе. Определяются необходимые требования к общему числу вычислителей в системе при заданном допустимом числе неисправных вычислителей. Предлагается алгоритм взаимной синхронизации, предшествующей выполнению согласования. Однако научно доказана корректность алгоритма только для неисправности одного вычислителя и при наличии определенной синхронности во взаимном обмене сообщениями (что объясняется достаточно ранней работой автора) применительно к имеющимся на тот момент структурам РМВС.

**Современные области применения синхронизации. Многокластерные и многокомплексные системы.** В современном мире распределенные системы становятся настолько масштабными и многозадачными, что их архитектура приобретает сложную, иногда иерархическую структуру, поэтому интерес к вопросам синхронизации еще более возрастает в связи с увеличением ее актуальности. Кроме того, появляются новые методы синхронизации часов в РМВС, ориентированные на современные технологии и области использования.

Авторы [14] считают, что вычислительные узлы должны быть сгруппированы в кластеры с использованием функциональных критериев, как это было предложено в [15].

Множество связанных между собой кластеров называется многокластерной системой. Согласно [14], существует три причины, по которым компоненты многокластерных систем нуждаются в едином понятии времени [16]:

1) для обеспечения осмысленного обмена наблюдениями через интерфейсы, поскольку наблюдения основаны на времени;

2) время имеет фундаментальное значение для распределенных систем управления, так как оно дает возможность коллективно выполнять вычислительные действия для управления объектом или процессом;

3) общее понятие времени является ключом к составу и надежности таких систем.

Упомянутые выше многочисленные работы свидетельствуют о том, что эта область исследований довольно хорошо освещена в публикациях [2, 3, 5, 17–25]. Но при синхронизации часов в больших распределенных системах требуются новые подходы к решению этой проблемы. Тем не менее ни при одном из подходов не используются преимущества таких аппаратных свойств часов, как систематическая скорость дрейфа.

Цель работы [14] — представить алгоритм, позволяющий осуществлять синхронизацию мультикластерных систем. В этой работе основное внимание уделяется повышению качества синхронизации за счет различной обработки систематических и стохастических типов ошибок.

Мультикластерная синхронизация достигается путем выполнения внешней синхронизации часов (External Clock Synchronization, ECS) глобального времени отдельных кластеров с применением реляционных часов в качестве эталонов. Реляционные часы — это виртуальные часы (другого кластера, отличного от внешнего синхронизирующего кластера), которые используются в качестве ссылки для ECS. Реляционные часы не являются идеальными представлениями реального времени, имеют множество скоростей дрейфа, отличающихся от 0, и точность по отношению к стандарту времени (такому, как глобальная система позиционирования), также отличную от 0. Узел, который может считывать время реляционных часов в дополнение к своему виртуальному времени, называется главным узлом времени. Главный узел времени может вычислить разницу между своим виртуальным временем и реляционным временем. Эта разница называется смещением и обозначается  $\Omega_i^j$  для главного узла  $j$  в момент времени  $t$ .

Информация о времени передается от одного кластера к другому через специальные компоненты — шлюзы, каждый из которых состоит из двух узлов. Один узел является элементом одного кластера с виртуальными часами, внутренне синхронизированными с другими виртуальными часами этого кластера. Другой узел представляет собой элемент другого кластера и также имеет виртуальные часы, которые внутренне синхронизированы с виртуальными часами своего кластера. Виртуальные часы каждого узла содержат в себе представление глобального времени кластера.

Отказоустойчивость достигается посредством использования достаточного количества реляционных часов, которые доступны через главные временные узлы. Предполагается, что реляционные часы могут выходить из строя независимо друг от друга. Еще одним средством обеспечения отказоустойчивого поведения является самостабилизирующаяся конструкция алгоритма ECS. Многокластерная синхронизация часов достигается путем синхронизации глобального времени кластера с реляционными часами [14].

Применение встраиваемых систем также предполагает необходимость использования синхронизации часов [26], требуемой для учета изменений параметров объекта управления, ведения журналов событий, выполнения действий, связанных с формированием и использованием расписаний. Работа [26] посвящена исследованию

названных проблем во встраиваемых системах малой временной точности (МВТ).

Для встраиваемых систем критичными могут быть требования к таким параметрам как точность, надежность и безопасность; при этом увеличиваются их сложность, стоимость, ресурсоемкость. Для встраиваемых систем высокой и средней временной точности существует много решений проблемы обеспечения точного времени: специализированные серверы времени с соблюдением атомных стандартов точности, избыточность эталонных источников, сложные алгоритмы калибровки и синхронизации часов и многое другое. Но данные решения неприемлемы для применения во встраиваемых системах МВТ вследствие их дороговизны, сложности или недоступности для широкого круга разработчиков, требования высоких вычислительных мощностей от компонентов системы. При этом методы коррекции времени во встраиваемых системах МВТ требуют обеспечения необходимой стабильности времени, масштабируемости, пониженного энергопотребления.

В [27] описывается модель отказоустойчивой синхронизации часов в сенсорных сетях, адаптированная из [21], и проводится различие между реальным временем (*real time*) и тактовым (*clock time*). Реальное время — это предполагаемый ньютоновский временной фрейм, который нельзя непосредственно наблюдать, тактовое время можно наблюдать на часах сенсорных узлов. Часы  $C$  можно считать отображением реального времени в тактовое, т. е.  $T = C(t)$  — тактовое время в реальном времени  $t$ . Часы  $C$  считаются правильно работающими, если у них скорость дрейфа от реального времени ограничена константой  $\rho > 0$  для любых точек реального времени  $t_1$  и  $t_2$ , где  $t_1 < t_2$ :

$$\frac{t_2 - t_1}{1 + \rho} < C(t_2) - C(t_1) < (1 + \rho)(t_2 - t_1).$$

Скорость дрейфа между любыми двумя правильно работающими часами ограничена величиной  $\varepsilon = \rho(2 + \rho)/(1 + \rho)$ , которая меньше  $2\rho$  [27]. Сенсорные узлы обычно содержат недорогие кварцевые генераторы, а обычная скорость ухода часов составляет десятки микросекунд [28]. Узел датчика считается исправным, если он правильно выполняет заданный алгоритм синхронизации часов и его часы работают правильно.

Если узел выполняет настройку часов в момент времени  $t$ , то используются обозначения  $C(t)$  и  $C^+(t)$  для представления времени часов до и после их настройки соответственно.

Пусть существует верхняя граница задержки  $\psi$  для сообщения, которое должно быть отправлено узлом, передано и обработано получателями этого сообщения. Если узел  $i$  посылает сообщение в  $C_{p_i}(t_1)$ , узел  $j$  получает сообщение в  $C_{p_j}(t_2)$ , где  $0 < t_2 - t_1 < \psi$ , и узел  $j$  настраивает свои часы следующим образом [27]:

$$C_{p_j}^+(t_2) = C_{p_i}(t_1),$$

то  $C_{p_i}(t_2) - C_{p_j}^+(t_2) = C_{p_i}(t_2) - C_{p_i}(t_1) < \phi$ , где  $\phi = (1 + \rho)\psi$ .

В [27] рассматриваются также вопросы, связанные с отказоустойчивой синхронизацией часов в кластере:

- широковещательная аутентификация;
- отказоустойчивый алгоритм синхронизации часов;
- кластеризация;
- глобальная синхронизация часов;
- сравнение с предыдущими методиками.

Автор [29] рассматривает две задачи:

1) децентрализованное выделение в распределенной системе цифровых устройств (ЦУ) лидера, который будет управлять синхронизацией остальных, ведомых ЦУ, посылая им сообщения-команды. В ответ на получение команды, ведомые ЦУ-источники должны отправить через средства связи свои частные сообщения так, чтобы они поступили на все ЦУ-приемники как общее сообщение, в котором частные сообщения занимают заранее заданные лидером места;

2) в ответ на команду лидера ведомые ЦУ должны выполнить в заданные лидером моменты времени действия, локализованные в пределах ЦУ, т. е. не требующие использования сетевого взаимодействия между ЦУ.

В результате автор [29] отмечает следующее: синхронизация основана на взаимодействии устройств трех видов:

- 1) группы ведомых цифровых устройств;
- 2) выделяемого из их состава лидера, управляющего порядком действий цифровых устройств;
- 3) центра — ретранслятора сигналов.

Сигналы, направляемые устройствами в центр, могут поступать также к приемникам устройств, накладываясь друг на друга, что не мешает взаимодействию устройств, так как приемники не воспринимают эти сигналы. Центр принимает сигналы цифровых устройств и каждый из этих сигналов заменяет качественно отличающимся от него собственным сигналом, одновременно направляемым всем устройствам. Лидер регулирует передачу сигналов цифровыми устройствами так, чтобы сигналы поступали в центр в требуемое лидеру время.

В результате отдельные сообщения многих устройств объединяются в общее сообщение, передаваемое центром всем устройствам по общему каналу с высокой скоростью, не зависящей от размеров системы устройств. Это уменьшает нагрузку на канал связи и ускоряет выполнение операций вида «все — всем», «все — одному», «один — всем». В частности, таким образом целесообразно осуществлять связь с интеллектуальными периферийными устройствами, в том числе с подвижными.

Работа [29] представляет несомненный интерес, однако использовать методы, предлагаемые автором при разработке сбое- и отказоустойчивых РМВС ответственного применения, не представляется возможным вследствие нарушения в определенные моменты принципа построения  $p-2-p$ , который является одним из наиболее важных факторов повышения живучести таких систем.

Современные автоматические системы управления технологическими процессами (АСУ ТП) представляют собой сложные программно-технические комплексы, состоящие из множества вычислительных компонентов, объединенных вычислительной сетью. Дополнительной характеристикой современных АСУ ТП является включение АСУ ТП объекта в глобальную информационную структуру в рамках некоторого объединения. Высокие требования, предъявляемые к достоверности и анализируемости обрабатываемой в системе информации, обеспечиваются как синхронностью работы различных компонентов системы, так и возможностью соотнесения ее временной шкалы с абсолютной шкалой времени.

В [30] рассмотрена проблема синхронизации часов в распределенной компьютерной управляющей системе, функционирующей в рамках АСУ ТП объекта. В качестве основного инструмента, обеспечивающего синхронизацию часов, исследуется система протокола сетевого времени (Network Time Protocol, NTP) [31], приведена ее общая математическая модель, обсуждается влияние параметров модели на характеристики NTP. Кроме того показана возможность в качестве альтернативного реализованному в NTP критерию оптимизации использовать при выборе параметров функционал квадратичной ошибки. Исследование выполнено при частичной финансовой поддержке Российского фонда фундаментальных исследований в рамках научного проекта № 19-29-06044, гл. 3, что подтверждает научную и практическую значимость работы.

Вопросы синхронизации часов затрагивают различные области современной жизни: гражданские [32], военные [33], двойного назначения [34]. В частности, в [34] рассмотрена система синхронизации малобазовых пространственно-когерентных распределенных радиолокационных комплексов. Проведен анализ существующих на сегодняшний день методов синхронизации пространственно-распределенных систем.



**Заключение.** Одним из базовых механизмов [35] организации современных сбое- и отказоустойчивых РМВС ответственного применения является синхронизация часов. Начиная с работ [2, 36–39] актуальность вопросов синхронизации многократно возрастает с увеличением сложности архитектуры РМВС. Анализ публикаций позволяет сделать следующие выводы.

Аналогично автору работы [40], будем считать априори, что существующее многообразие методов синхронизации позволяет выбрать оптимальный для конкретной архитектуры и задач РМВС.

В процессе работы РМВС как в режиме тестирования и диагностирования, так и при выполнении целевых задач, возможно возникновение нештатных ситуаций, которые способны создать проблемы при синхронизации. Наиболее опасная из них — возникновение ошибки в момент синхронизации часов, что может нарушить сам процесс синхронизации. Тогда перестает выполняться один из базовых механизмов [41] организации сбое- и отказоустойчивости в МВС, и дальнейшая работа РМВС становится невозможной.

Иногда требуется выполнить обратную задачу: рассинхронизацию, чтобы синфазная помеха не привела к одновременной ошибке в каналах резервирования. В этом случае можно говорить о синхронизации принятия решения в условиях асинхронных частных вычислений в каналах резервирования.

Наиболее подходящим методом синхронизации для современных сбое- и отказоустойчивых многокомплексных РМВС ответственного применения является метод единого системного времени, возможно, с различными модификациями, учитывающими их многозадачность и кратность неисправностей. Синхронизация должна выполняться с точностью до раунда обмена сообщениями в полносвязных системах или до кванта [41, 42] для неполносвязных систем.

## ЛИТЕРАТУРА

- [1] Ашарина И.В. Синхронизация часов в распределенных многомашиных вычислительных системах. Часть I. *Инженерный журнал: наука и инновации*, 2021, вып. 4. <http://dx.doi.org/10.18698/2308-6033-2021-4-2074>
- [2] Lamport L., Melliar-Smith P.M. Synchronizing clocks in the presence of faults. *Journal of the ACM*, 1985, vol. 32, no. 1, pp. 52–78.
- [3] Shin Kang G., Ramanathan P. Clock synchronization of a large multiprocessor system in the presence of malicious faults. *IEEE transactions on computers*, 1987, vol. C-36, no. 1, pp. 2–12.
- [4] Dolev D., Halpern J.Y., Strong R. On the possibility and impossibility of achieving clock synchronization. *Proceedings of the 16th Annual ACM STOC. April 30 — May 2, 1984, Washington D.C., USA*. New York, ACM, 1984, pp. 504–511.
- [5] Lundelius-Welch J., Lynch N. A new fault-tolerant algorithm for clock synchronization. *Information and Computation*, 1988, no. 77, pp. 1–36. DOI: 10.1016/0890-5401(88)90043-0

- [6] Halpern J., Simons B., Strong R. An efficient fault-tolerant algorithm for clock synchronization. *Proceedings of the 3rd annual ACM symposium on Principles of distributed computing, August 27–29, 1984, Vancouver, B. C., Canada*. Association for Computing Machinery, New York, 1984, pp. 75–88. DOI: 10.1145/800222.806738
- [7] Mahaney S., Schneider F. Inexact agreement: Accuracy, precision and graceful degradation. *Proceedings of 4th International Symposium on Principles of Distributed Computing, August, 5–7, 1985, Minaki, Ontario, Canada*. New York, Association for Computing Machinery, 1985, pp. 237–249. DOI: 10.1145/323596.323618
- [8] Drummond R. *Impact of communication networks on fault-tolerant distributed computing*. Ph.D. dissertation thesis. Ithaca, N.Y., Department of Computer Science, Cornell University, 1986, 95 p.
- [9] Marzullo K. *Maintaining the time in a distributed system. An example of a loosely-coupled distributed service*. Ph.D. dissertation. Stanford, California, Department of Electrical Engineering, Stanford University, 1984, 128 p.
- [10] Srikanth T.K., Toueg S. Optimal Clock Synchronization. *Journal of the Association for Computing Machinery*, 1987, vol. 34, no. 3, pp. 626–645.
- [11] Diffie W., Hellman M. New directions in cryptography. *IEEE Trans. Inf. Theory*, 1976, no. 22, pp. 644–654.
- [12] Srikanth T.K., Toueg S. Simulating authenticated broadcasts to derive simple fault-tolerant algorithms. *Technical Report*. Ithaca, N.Y., Cornell University, Department of Computer Science, 1984, pp. 84–623.
- [13] Лобанов А.В. Синхронизация и взаимное информационное согласование в распределенных вычислительных системах. *Программирование*, 1997, № 2, с. 76–80.
- [14] Paulitsch M., Steiner W. Fault-tolerant clock synchronization for embedded distributed multi-cluster systems. *Conference: Real-Time Systems, Proceedings. 15th Euromicro Conference, August 2003*. Technische Universität Wien, Vienna, Austria. DOI: 10.1109/EMRTS.2003.1212750
- [15] Phillips A. Functional decomposition in a vehicle control system. *American Control Conference, Anchorage, AK, May 8–10, 2002*. Anchorage, AK, USA, IEEE, vol. 5, pp. 3713–3718. DOI: 10.1109/ACC.2002.1024505
- [16] Kopetz H., Bauer G. The time-triggered architecture. *Proceedings of the IEEE*, 2003, vol. 91, iss. 1, pp. 112–126. DOI: 10.1109/JPROC.2002.805821
- [17] Kopetz H., Ochsenreiter W. Clock synchronization in distributed real-time computer systems. *IEEE Transactions on Computers*, 1987, C-36, no. 8, pp. 933–940.
- [18] Srikanth T.K., Toueg S. Optimal clock synchronization. *Journal of the Association for Computing Machinery*, 1987, vol. 34, no. 3, pp. 626–645.
- [19] Verissimo P., Casimiro A., Rodrigues L. Cesiumspray: a precise and accurate global time service for large scale systems. *Journal of Real-Time Systems*, 1997, vol. 12, no. 3, pp. 243–294. DOI: 10.1023/A:1007949113722
- [20] Pfluegl M., Blough D. A new and improved algorithm for fault-tolerant clock synchronization. *Journal of Parallel and Distributed Computing*, 1995, vol. 27, pp. 1–14.
- [21] Dolev D., Halpern J., Simons B., Strong R. Dynamic fault-tolerant clock synchronization. *Journal of the Association for Computing Machinery*, 1995, vol. 42, no. 1, pp. 143–185.
- [22] Fetzer C., Cristian F. Integrating external and internal clock synchronization. *Journal of Real-Time Systems*, 1997, vol. 12, no. 2, pp. 123–172.
- [23] Schmid U., Schossmaier K. Interval-based clock synchronization. *Real-Time Systems*, 1997, vol. 12, pp. 173–228.

- [24] De Azevedo M., Blough D. Fault-tolerant clock synchronization of large multi-computers via multistep interactive convergence. *Proceedings of 16th International Conference on Distributed Computing Systems, Hong Kong, China, May 27–30, 1996*. New York, IEEE, 1996, pp. 249–257.  
DOI: 10.1109/ICDCS.1996.507923
- [25] Kopetz H., Krüger A., Millinger D., Schedl A. A synchronization strategy for a time-triggered multicluster real-time system. *Proceedings of the 14th Symposium on Reliable Distributed Systems, September 13–15, 1995, Bad Neuenahr, Germany*. IEEE Computer Society Press, 1995, pp. 154–161.  
DOI: 10.1109/RELDIS.1995.526223
- [26] Ковязина Д.Р. Автоматическая калибровка и синхронизация времени во встраиваемых вычислительных системах. *Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики*, 2009, № 2 (60), с. 87–93.
- [27] Sun K., Ning P., Wang C. Fault-tolerant cluster-wise clock synchronization for wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 2005, vol. 2, iss. 3, pp. 177–189. DOI: 10.1109/TDSC.2005.36
- [28] Romer K., Blum P., Meier L. Time synchronization and calibration in wireless sensor networks. *Wireless Sensor Networks*. Ivan Stojmenovic, ed. Hoboken, NJ, Wiley-Interscience, 2005, pp.199–237.
- [29] Стецюра Г.Г. Синхронизация взаимодействия цифровых устройств с помощью центра ретрансляции сигналов. *Автоматика и телемеханика*, 2012, вып. 5, с. 111–124.
- [30] Промыслов В.Г., Семенков К.В., Тимофеев М.Ю. *Проблемы синхронизации времени в распределенных автоматизированных системах управления технологическими процессами*.  
URL: <https://mlsd2020.ipu.ru/proceedings/1431-1437.pdf> (дата обращения 15.03.2020).
- [31] Mills D.L. Internet time synchronization: the network time protocol. *IEEE Transactions on Computers*, 1991, vol. 39, no. 10, pp. 1482–1493.
- [32] Mills D., Martin J., Burbank J., Kasch W. Network time protocol version 4: protocol and algorithms specification, RFC5905 (Standard). *Internet Engineering Task Force (IETF), Technical report*, 2010, pp. 68–88.  
URL: <https://tools.ietf.org/html/rfc5905> (дата обращения 15.03.2017).
- [33] Беляков М.О., Борисов Д.А., Селезнев А.А. Синхронизация времени внутри группировки ПВО с использованием проводных и радиоканалов связи. *Вопросы радиоэлектроники*, 2017, № 12, с. 36–43.
- [34] Крючков И.В., Филатов А.А. Синхронизация подвижных модулей распределенных радиолокационных комплексов. *Инженерный журнал: наука и инновации*, 2012, вып. 8, с. 45–52. DOI: 10.18698/2308-6033-2012-8-316
- [35] Лобанов А.В., Сиренко В.Г. Проблема отказоустойчивости в сетевых информационно-управляющих системах. *Образовательные ресурсы и технологии*, 2014, № 2 (5), с. 115–121.
- [36] Lamport L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 1978, vol. 21, no. 7, pp. 558–565.
- [37] Lamport L., Shostak R., Pease M. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 1982, vol. 4, no. 3, pp. 382–401.
- [38] Lamport L. The implementation of reliable distributed multiprocess systems. *Computer Network*, 1978, vol. 2, pp. 95–114.
- [39] Lamport L. Using time instead of timeout for fault-tolerant distributed systems. *ACM Transactions on Programming Languages and Systems*, 1984, vol. 6, no. 2, pp. 254–280. DOI: 10.1145/2993.2994

- [40] Liskov B. Practical uses of synchronized clocks in distributed systems. *Distributed Computing, Periodicals Distributed Computing*, 1993, vol. 6, no. 4, pp. 211–219. DOI: 10.1007/BF02242709
- [41] Лобанов А.В. Модели замкнутых многомашинных вычислительных систем со сбое- и отказоустойчивостью на основе репликации задач в условиях возникновения враждебных неисправностей. *Автоматика и телемеханика*, 2009, № 2, с. 171–189.
- [42] Ашарина И.В., Лобанов А.В., Мищенко И.Г. Взаимное информационное согласование в неполносвязных многомашинных вычислительных системах. *Автоматика и телемеханика*, 2003, № 5, с. 190–198.

Статья поступила в редакцию 17.03.2021

Ссылку на эту статью просим оформлять следующим образом:

Ашарина И.В. Синхронизация часов в распределенных многомашинных вычислительных системах. Часть II. *Инженерный журнал: наука и инновации*, 2021, вып. 5. <http://dx.doi.org/10.18698/2308-6033-2020-5-2081>

**Ашарина Ирина Владимировна** — канд. техн. наук, доцент, старший научный сотрудник АО «НИИ «Субмикрон». e-mail: [asharinairina@mail.ru](mailto:asharinairina@mail.ru)

## Clock synchronization in distributed multicomputer systems. Part II

© I.V. Asharina

JSC Research Institute Submicron, Moscow, Zelenograd, 124460, Russia

*The study substantiates the necessity of clock synchronization in distributed multicomputer systems. The basic definitions related to the concept of clock synchronization are given, and methods of clock synchronization are classified. Increasing the lifecycle of failure- and fault-tolerant distributed multicomputer systems for critical application is one of the most urgent problems at the current level of technology development. This is especially true for unattended distributed multicomputer systems for space applications. The second part deals with synchronization in systems with Byzantine faults, and this is a complex task due to the characteristic features of the fault model. The synchronization process in multi-cluster and multi-complex systems is associated with the multitasking of such systems, which makes the synchronization process even more relevant and multi-criteria. The paper considers the modern technologies providing the synchronization process in systems of critical use.*

**Keywords:** distributed multicomputer system, failure and fault tolerance, clock synchronization, dynamic redundancy, Byzantine fault

### REFERENCES

- [1] Asharina I.V. *Inzhenerny zhurnal: nauka i innovatsii — Engineering Journal: Science and Innovation*, 2021, iss. 4. DOI: 10.18698/2308-6033-2021-4-2074
- [2] Lamport L., Melliar-Smith P.M. Synchronizing clocks in the presence of faults. *J. Assoc. Comput. Mach.*, 1985, vol. 32 (1), pp. 52–78.
- [3] Shin Kang G., Ramanathan P. Clock synchronization of a large multiprocessor system in the presence of malicious faults. *IEEE Transactions on Computers*, 1987, vol. C-36, no. 1, pp. 2–12.
- [4] Dolev D., Halpern J.Y., Strong R. On the possibility and impossibility of achieving clock synchronization. *Proceedings of the 16th Annual ACM STOC. April 30 — May 2, 1984, Washington D.C., USA*. New York, ACM, 1984, pp. 504–511.
- [5] Lundelius-Welch J., Lynch N. A new fault-tolerant algorithm for clock synchronization. *Information and Computation*, 1988, no. 77, pp. 1–36. DOI: 10.1016/0890-5401(88)90043-0
- [6] Halpern J., Simons B., Strong R. An efficient fault-tolerant algorithm for clock synchronization. *Proceedings of the 3rd annual ACM symposium on Principles of distributed computing, August 27–29, 1984, Vancouver, B.C., Canada*. New York, Association for Computing Machinery, 1984, pp. 75–88. DOI: 10.1145/800222.806738
- [7] Mahaney S., Schneider F. Inexact agreement: Accuracy, precision and graceful degradation. *Proceedings of 4th International Symposium on Principles of Distributed Computing, August, 5–7, 1985, Minaki, Ontario, Canada*. New York, Association for Computing Machinery, 1985, pp. 237–249. DOI: 10.1145/323596.323618
- [8] Drummond R. *Impact of communication networks on fault-tolerant distributed computing. Ph.D. dissertation thesis*. Ithaca, N.Y., Department of Computer Science, Cornell University, 1986, 95 p.

- [9] Marzullo K. *Maintaining the time in a distributed system. An example of a loosely-coupled distributed service. Ph.D. dissertation.* Stanford, California, Department of Electrical Engineering, Stanford University, 1984, 128 p.
- [10] Srikanth T.K., Toueg S. Optimal clock synchronization. *Journal of the Association for Computing Machinery*, 1987, vol. 34, no. 3, pp. 626–645.
- [11] Diffie W., Hellman M. New directions in cryptography. *IEEE Trans. Inf. Theory IT*, 1976, no. 22, pp. 644–654.
- [12] Srikanth T.K., Toueg S. Simulating authenticated broadcasts to derive simple fault-tolerant algorithms. Tech. Rep. 84-623, *Dept. of Computer Science*, Cornell Univ., Ithaca, N.Y., July 1984. (Also to appear in *Distributed Computing*, Springer-Verlag, New York.)
- [13] Lobanov A.V. *Programmirovaniye — Programming and Computer Software*, 1997, no. 2, pp. 76–80.
- [14] Paulitsch M., Steiner W. Fault-tolerant clock synchronization for embedded distributed multi-cluster systems. *Conference: Real-Time Systems, Proceedings. 15th Euromicro Conference*, August 2003. Technische Universität Wien, Vienna, Austria. DOI: 10.1109/EMRTS.2003.1212750
- [15] Phillips A. Functional decomposition in a vehicle control system. *American Control Conference*, Anchorage, AK, May 8–10, 2002. Anchorage, AK, USA, IEEE, vol. 5, pp. 3713–3718. DOI: 10.1109/ACC.2002.1024505
- [16] Kopetz H., Bauer G. The time-triggered architecture. *Proceedings of the IEEE*, 2003, vol. 91, iss. 1, pp. 112–126. DOI: 10.1109/JPROC.2002.805821
- [17] Kopetz H., Ochsenreiter W. Clock synchronization in distributed real-time computer systems. *IEEE Transactions on Computers*, 1987, C-36, no. 8, pp. 933–940.
- [18] Srikanth T.K., Toueg S. Optimal clock synchronization. *Journal of the Association for Computing Machinery*, 1987, vol. 34, no. 3, pp. 626–645.
- [19] Verissimo P., Casimiro A., Rodrigues L. Cesium spray: a precise and accurate global time service for large scale systems. *Journal of Real-Time Systems*, 1997, vol. 12, no. 3, pp. 243–294. DOI: 10.1023/A:1007949113722
- [20] Pfluegl M., Blough D. A new and improved algorithm for fault-tolerant clock synchronization. *Journal of Parallel and Distributed Computing*, 1995, vol. 27, pp. 1–14.
- [21] Dolev D., Halpern J., Simons B., Strong R. Dynamic fault-tolerant clock synchronization. *Journal of the Association for Computing Machinery*, 1995, vol. 42, no. 1, pp. 143–185.
- [22] Fetzer C., Cristian F. Integrating external and internal clock synchronization. *Journal of Real-Time Systems*, 1997, vol. 12, no. 2, pp. 123–172.
- [23] Schmid U., Schossmaier K. Interval-based clock synchronization. *Real-Time Systems*, 1997, vol. 12, pp. 173–228.
- [24] De Azevedo M., Blough D. Fault-tolerant clock synchronization of large multicomputers via multistep interactive convergence. *Proceedings of 16th International Conference on Distributed Computing Systems, Hong Kong, China, May 27–30, 1996.* New York, IEEE, 1996, pp. 249–257. DOI: 10.1109/ICDCS.1996.507923
- [25] Kopetz H., Krüger A., Millinger D., Schedl A. A synchronization strategy for a time-triggered multicluster real-time system. *Proceedings of the 14th Symposium on Reliable Distributed Systems, September 13–15, 1995, Bad Neuenahr, Germany.* IEEE Computer Society Press, 1995, pp. 154–161. DOI: 10.1109/RELDIS.1995.526223
- [26] Kovyazina D.R. *Nauchno-tehnicheskiy vestnik Sankt-Peterburgskogo gosudarstvennogo universiteta informatsionnykh tekhnologii, mekhaniki i optiki — Sci-*

- entific and Technical Journal of Information Technologies, Mechanics and Optics*, 2009, no. 2 (60), pp. 87–93.
- [27] Sun K., Ning P., Wang C. Fault-tolerant cluster-wise clock synchronization for wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 2005, vol. 2, iss. 3, pp. 177–189. DOI: 10.1109/TDSC.2005.36
- [28] Romer K., Blum P., Meier L. Time synchronization and calibration in wireless sensor networks. *Wireless Sensor Networks*, Ivan Stojmenovic, ed. Hoboken, NJ, Wiley-Interscience, 2005, pp.199–237.
- [29] Stetsyura G.G. *Avtomatika i telemekhanika — Automation and Remote Control*, 2012, no. 5, pp. 111–124.
- [30] Promyslov V.G., Semenov K.V., Timofeev M.Yu. *Problemy sinkhronizatsii vremeni v raspredelennykh avtomatizirovannykh sistemakh upravleniya tekhnologicheskimi protsessami* [Time synchronization problems in distributed automated process control systems]. Available at: <https://mlsd2020.ipu.ru/proceedings/1431-1437.pdf>
- [31] Mills D.L. Internet time synchronization: the network time protocol. *IEEE Transactions on Computers*, 1991, vol. 39, no. 10, pp. 1482–1493.
- [32] Mills D., Martin J., Burbank J., Kasch W. Network time protocol version 4: protocol and algorithms specification, RFC5905 (Standard). *Internet Engineering Task Force (IETF), Technical report*, 2010, pp. 68–88. Available at: <https://tools.ietf.org/html/rfc5905> (accessed March 15, 2017).
- [33] Beliakov M.O., Borisov D.A., Seleznev A.A. *Voprosy radioelektroniki — Issues of radio electronics*, 2017, no. 12, pp. 36–43.
- [34] Kryuchkov I.V., Filatov A.A. *Inzhenerny zhurnal: nauka i innovatsii — Engineering Journal: Science and Innovation*, 2012, iss. 8, pp. 45–52. DOI: 10.18698/2308-6033-2012-8-316
- [35] Lobanov A.V., Sirenko V.G. *Obrazovatelnye resursy i tekhnologii (Educational resources and technologies)*, 2014, no. 2 (5), pp. 115–121.
- [36] Lamport L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 1978, vol. 21, no. 7, pp. 558–565.
- [37] Lamport L., Shostak R., Pease M. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 1982, vol. 4, no. 3, pp. 382–401.
- [38] Lamport L. The implementation of reliable distributed multiprocess systems. *Computer Network*, 1978, vol. 2, pp. 95–114.
- [39] Lamport L. Using time instead of timeout for fault-tolerant distributed systems. *ACM Transactions on Programming Languages and Systems*, 1984, vol. 6, no. 2, pp. 254–280. DOI: 10.1145/2993.2994
- [40] Liskov B. Practical uses of synchronized clocks in distributed systems. *Distributed Computing, Periodicals Distributed Computing*, 1993, vol. 6, no. 4, pp. 211–219. DOI: 10.1007/BF02242709
- [41] Lobanov A.V. *Avtomatika i telemekhanika — Automation and Remote Control*, 2009, no. 2, pp. 171–189.
- [42] Asharina I.V., Lobanov A.V., Mischenko I.G. *Avtomatika i telemekhanika — Automation and Remote Control*, 2003, no. 5, pp. 190–198.

**Asharina I.V.**, Cand. Sc. (Eng.), Assoc. Professor, Senior Research Fellow, JSC Research Institute Submicron. e-mail: [asharinairina@mail.ru](mailto:asharinairina@mail.ru)