ИПМ им.М.В.Келдыша РАН

Абрау-2019 • Труды конференции

О. Панарин, И. Захаров

Особенности мониторинга мобильных систем обработки информации

Рекомендуемая форма библиографической ссылки

Панарин О., Захаров И. Особенности мониторинга мобильных систем обработки информации // Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции (23-28 сентября 2019 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2019. — С. 551-560. — URL: http://keldysh.ru/abrau/2019/theses/81.pdf doi:10.20948/abrau-2019-81

Размещена также презентация к докладу

Особенности мониторинга мобильных систем обработки информации

О. Панарин[1], И. Захаров[1]

[1] Сколковский институт науки и технологий
Территория Инновационного Центра "Сколково", улица Нобеля, д. 3
Москва 121205 Россия

Аннотация. Рассматривается реализация подсистемы мониторинга систем обработки информации на мобильных платформах и ее применение на беспилотных автомобилях. В условиях беспилотной эксплуатации автомобиля предъявляются наиболее жёсткие требования к надежности систем обработки информации, принятию решения о готовности этих систем к эксплуатации и обеспечению анализа их возможных сбоев. Представленная система мониторинга pLOG сочетает в себе функционал записи событий операционной системы устройств и измерений параметров систем в реальном времени, при этом запись производится как файлы, так и в базу данных временных рядов (TSDB). При этом каждый сервер в составе системы обработки информации на мобильных платформах дублирует запись обо всех событиях в системе.

Ключевые слова: сенсорные данные, распределенные системы, мониторинг

Monitoring mobile information processing systems

O. Panarin[1], I. Zacharov[1]

[1] Skolkovo Institute of Science and Technology
Skolkovo Innovation Center, 3 Nobel Street, Moscow 121205, Russia

Abstract. We describe the implementation of the monitoring for the IT systems at the core of the autonomous driving vehicle. The role of the monitoring is to assist in decision to start the driving cycle and continuous assessment for the fitness to drive the vehicle. The requirements for the monitoring system with the increased resiliency and data replication make it sufficiently different from standard monitoring systems and warrant a unique implementation tuned for the autonomous driving requirements. The monitoring system combines the OS events and real-time measurements of sensor data. The information is stored in flat files for emergency access as well as in a Time Series Data Base (TSDB).

Keywords: mobile systems, distributed systems, sensor data, monitoring

Введение

Автоматическое управление беспилотным автомобилем осуществляется системой обработки информации с множества датчиков (например, дальномеров – LIDAR, системы стерео-зрения, систем глобального позиционирования и т.д.) в комбинации со встроенными картами, классификацией объектов в реальном времени и принятием решений по управлению (нейронные сети, экспертные системы). Система обработки информации строится на основе современных высокопроизводительных систем в усиленном корпусе с водяным охлаждением. Совокупная производительность устанавливаемых систем обработки информации в беспилотном автомобиле достигает порядка 100 ТФлоп/с (на операциях с плавающей точкой одинарной точности), а потребляемая мощность до 5 кВт на современном витке технологического развития. Вычислительная мощность распределяется по нескольким системам, обычно привязанная к конкретным источникам информации, которые система обрабатывает. Например, система для обработки сигналов от дальномера, система для обработки сигналов от стереозрения левого крыла и т.п., а также система, объединяющая распознанные объекты и примающая решение по управлению.

Несмотря на специфическую реализацию в автомобиле, основа всех систем обработки информации – сервер на операционной системе Линукс с графическими картами (ГПУ) для ускорения вычислений. Вся система обработки информации для автономного управления автомобилем – это классический вычислительный кластер объединенный рядом сетей, в том числе высокопроизводительной сетью для сбора данных о работе всех подсистем в реальном времени и их запись со скоростью порядка 5 ГБ/с. Указанные параметры производительности необходимы и достаточны (при современном уровне технологий) для того, что бы с одной стороны обеспечивать управление автомобилем, с другой сохранять все данные для последующего моделирования работы автопилота в лаборатории (off line).

Другая система записи событий в системе беспилотного автомобиля – это запись информации, необходимой и достаточной для анализа возможных ДТП. Требование к этой подсистеме – безусловная сохранность данных, которые могут потребоваться для анализа действий системы с правовой точки зрения. Скорость записи и размер хранилища здесь не является определяющим фактором, достаточно сохранять данные за последние 30 секунд движения автомобиля до ДТП. Эта подсистема является обязательной для уровня 3 SAE и выше [1] для обеспечения страховых обязательств.

Темой данной статьи является подсистема мониторинга работы самих серверов и объединяющих их сетей. Перед началом движения нужно убедится, что все системы работают в штатном режиме, данные от датчиков мониторинга и состояния собираются, сервера загружены и готовы к работе и машина готова к эксплуатации. В процессе работы системы при эксплуатации

автомобиля сообщение о внештатных ситуациях должно выводится (передаваться) оператору автомобиля для восстановления работоспособности (уровень 2, 3 SAE) или приводить к остановке автомобиля (уровень 4, 5 SAE). При этом сам процесс мониторинга в штатном режиме не является критичной операцией и необходим только при возникновении сбоев в системе. В случае сбоя в серверах обработки данных накопленная информация должна быть достаточна для анализа ситуации.

Мониторинг серверных вычислительных кластеров является классической и успешно решённой задачей в центрах обработки данных (ЦОД) со множеством различных реализаций. Например, можно упомянуть Nagios [2] или Zabbix [3] как наиболее распространённые и наиболее гибкие системы мониторинга для ЦОД-ов. Для мониторинга кластера в автомобиле решаются похожие задачи, но имеющие существенные особенности.

Перечислим особенности, которые определяют необходимость и предпосылки разработки уникальной системы мониторинга беспилотного автомобиля:

- требование к надежности системы мониторинга как таковой – система должна допускать множественные точки отказа. Конечно, при единичном отказе уже принимается решение остановить автомобиль. Но для анализа возникшей ситуации система должна продолжать работу, чтобы позволить оператору “в поле” восстановить функционал как можно скорее и в полном объеме;
- Наличие нестандартных датчиков в системе. В широко известных системах мониторинга возможно подключение датчиков по стандарту IMPI и SNMP; в автомобиле есть свои стандарты (COM-Bus, сети типа 100Base-T1), требующие серьезной адаптации “стандартной” системы мониторинга;
- ограничение по количеству узлов кластера. Системы мониторинга для кластеров в ЦОД-е должны масштабироваться на сотни и тысячи узлов, в то время как система мониторинга кластера в автомобиле ограничена всего порядка 10-ю узлами. Эта особенность позволяет реализовывать системы с меньшей алгоритмической сложностью и соответственно увеличенной надежностью;
- отсутствием необходимости сохранять данные мониторинга на долгий срок. Данные мониторинга могут ограничиваться несколькими прогонными циклами автомобиля. Эта особенность позволяет уменьшить требуемое дисковое пространство и строить алгоритм надежности на основе полной дубликации приходящих данных на всех узлах;
- необходимость предоставления данных мониторинга через разные системы доступа, как WEB-интерфейс для более удобного анализа, так и

наиболее примитивный вход на систему без графической карты. Такой функционал необходим для штатного анализа и для восстановления системы после сбоя;

- система не должна требовать дополнительных настроек и должна работать в автоматическом режиме, так как комплектов беспилотных машин большое количество, то невозможно на каждом осуществлять настройку и ручное сопровождение. Только в случае нештатной ситуации требуется анализ возникающих проблем.

В следующих разделах рассмотрен способ реализации системы мониторинга высокопроизводительного кластера на мобильной платформе, удовлетворяющей всем выше перечисленным требованиям, и представлены результаты работы такой системы. Предварительно проанализируем альтернативные стратегии реализации мониторинга.

Альтернативные стратегии реализации системы мониторинга

Исследуемая платформа — это высокопроизводительный кластер, отдельные узлы которого используют ОС Линукс. Для мониторинга работы таких узлов существуют много наработок. Например, top, htop, nmon, vmstat, iotop, iftop, netstat, bmon, glances и т.д. (см список [4]). Все утилиты используют статистику ядра Линукс, которая отдается через интерфейс (псевдо-) файловой системы /proc. Так статистика использования процессорного времени читается из файла /proc/stat, распределение памяти /proc/meminfo, сеть /proc/net/dev, дисководы /proc/diskstats и т.д. При каждом чтении такого файла драйвер ядра отдает внутренние переменные через этот интерфейс. Разница между всеми утилитами заключается только в том, как каждая из них структурирует и представляет эти данные и с какой скоростью утилиты их обновляют для пользователя.

Собственно, для всего кластера есть также много решений, позволяющих судить о его состоянии. Например, nagios, ganglia, zabbix, cacti и т.д. (см список [5]). Эти решения получают статистику из тех же файлов /proc, но предоставляют в удобной форме сравнительный анализ по всем узлам кластера и многие из них (на пример zabbix) дополнительно записывают статистику в базу данных.

Насколько нам известно, производители мобильных платформ используют в той или иной мере комбинацию из вышеперечисленных утилит для мониторинга. Это связано с тем, что современная форма реализации мобильных платформ с использованием Линуксовых кластеров для тестирования технологического функционала является промежуточной к промышленному использованию возможностей беспилотных автомобилей. Существующие Линукс-кластеры общего назначения должны преобразоваться в специальные изделия, подходящие для промышленной эксплуатации.

Поэтому инвестиции в переписывание общедоступных утилит могут показаться нецелесообразными в виду ожиданий будущих изменений в вычислительной базе.

Мы занимаем другую позицию. Мы считаем, что эволюция Линуксовых кластеров на мобильной платформе сводится к специализации функций (и связанной миниатюризации) с сохранением существующих интерфейсов. С другой стороны, задача мониторинга преобразуется в задачу управления живучестью системы (health management), для которой существующие утилиты не предназначены. Более того, специализация мобильной вычислительной платформы подразумевает интеграцию множества сенсоров, на основе которых и будет строится система управления живучестью. При сохранении Линуксовых интерфейсов, такие системы как, например, Zabbix дают возможность интеграции специализированных подсистем для мониторинга, но совокупная сложность всей программной обвязки увеличивается, а это неприемлемо. Таким образом, программное обеспечение на специализированных вычислительных системах будущих промышленных мобильных платформ также будет специализированным. Предлагаемая новая система мониторинга развивается именно в этом направлении.

В перспективе система мониторинга преобразованная и интегрированная в систему обеспечения живучести мобильной платформы повторит этапы развития систем обеспечения безопасности в пилотируемой авиации [6]. В авиации этот процесс еще тоже далек от завершения [7], но учитывая этот опыт мы намечаем соответствующие вехи развития мониторинга для автомобильных вычислительных систем.

Реализация системы мониторинга для кластера в автомобиле

Схема реализации системы мониторинга pLOG основана на полной симметрии узлов кластера независимо от их характеристик или приложения, которое узел поддерживает. Хотя, конечно, есть логически выделенный узел для сбора данных о работе всех подсистем в автомобиле и обращение к данным мониторинга будет в штатном режиме осуществляется именно через этот узел, но все другие узлы кластера точно также собирают и сохраняют те же данные о работе всего кластера. В случае сбоя маршрутизаторов и потери сети, данные по работе непосредственно до сбоя могут быть получены с любого узла, а по работе после сбоя - по каждому узлу по отдельности. В случае потери одного узла, на любом другом узле сохраняются данные которые упавший узел посылал до сбоя и они могут быть использованы для анализа.

Для анализа собранных данных привлекаются временные метки, записываемые вместе с данными мониторинга. Кластер синхронизируется по времени протоколом PTP (IEEE 1588 [8]) с разрешением (интервалом) около 1 микросекунды. В случае потери сети дрифт часов на узлах не считается

критичным для анализа собираемых данных, т.к. временной промежуток работы в таком нештатном режиме ограничен несколькими (десятками) минутами.

Функционал сбора информации в системе реализован с помощью без интерфейсных программ (daemons), запускаемых автоматически во время начала работы операционной системы на узлах. Схема взаимодействия программ на одном узле показана на рис. 1, и каждый узел реализует данную схему.

На каждом узле работает ОС Linux, Plog daemon, Plog-net daemon и экземпляр Akumuli. Базовый daemon Plog стартует первым при запуске ОС, инициализирует структуры в памяти и собирает данные с датчиков и от ОС для записи в структуру данных в памяти и в файл. Plog обновляет эти данные каждые 2 секунды (настраиваемый интервал) и посылает эти данные пакетом multicast если сеть доступна. После запуска Plog стартует Plog-net daemon, его функция принимать пакеты от других серверов в кластере и записывать данные в структуру данных в оперативной памяти и в файл для каждого удаленного сервера (см. ниже).

Такое разделение функционала обусловлено тем, что локальный мониторинг нужен и без работающей сети. Локальный мониторинг собирает данные как с самого сервера, так и с подключенных к нему внешних датчиков оценивающих состояние маршрутизаторов или других элементов сети. Если есть сбой в охлаждении или подаче питания на маршрутизатор, то он не запустится, и эта информация будет доступна на мониторинге узла, к которому подключены датчики маршрутизатора.

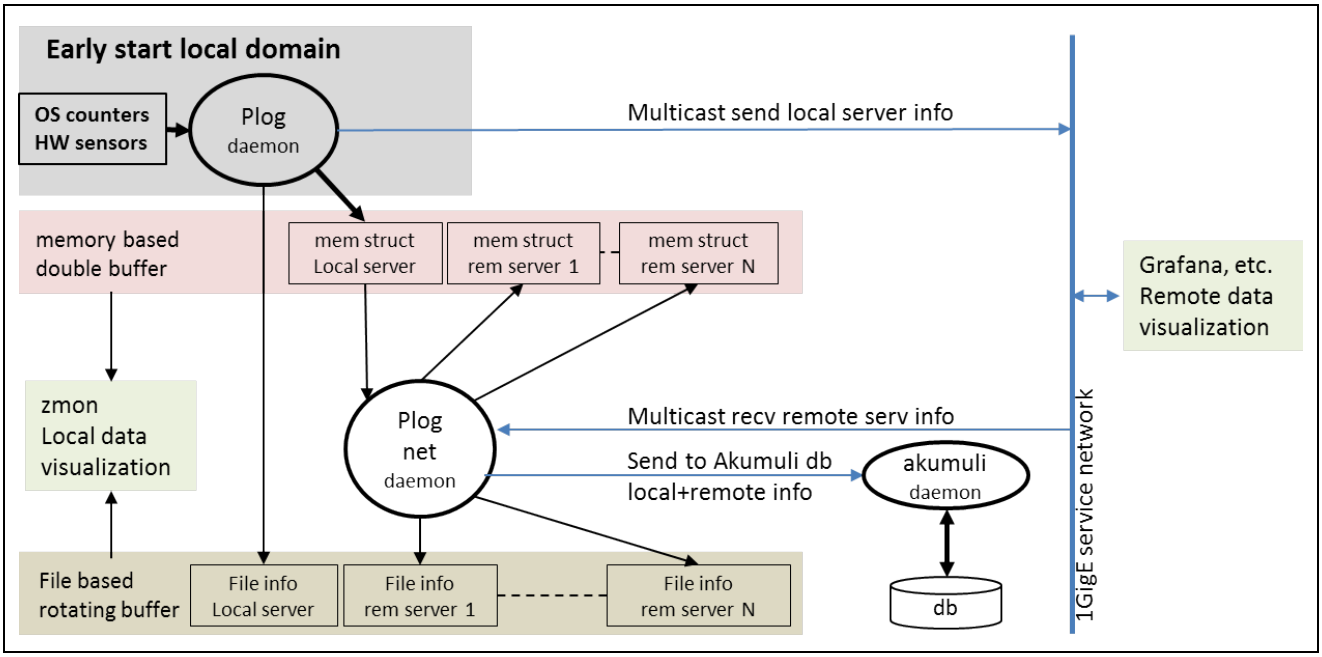

Рис. 1. Архитектура системы мониторинга кластера в автомобиле. Такая схема работы реализуется на каждом узле кластера.

Мониторинг узла и подключенных датчиков без работающей сети осуществляется командой zmon, которая выводит на консоль данные мониторинга. Команда написана в стиле известной утилиты nmon [4], используя библиотеку ncursors [9]. Команда zmon с флагом -R переключается на чтение информации, сохраненной в файлах и делает повторную прокрутку тех данных мониторинга, которые собирались в течении последних 20 минут (настраиваемый параметр).

Для обеспечения полного функционала мониторинга Plog-net daemon принимает пакеты от экземпляров Plog daemon на других узлах. Таким образом, на каждом узле собирается полная картина мониторинга кластера.

Так же, Plog-net daemon связывается с экземпляром агента базы данных Akumuli [10], который копит данные на среднесрочную перспективу и может отдавать данные по запросу в формате json на графический ресурс Grafana [11]. В принципе, один экземпляр Akumuli запущен на каждом узле кластера.

Это сделано для симметрии и не требует отдельной конфигурации кластерных ресурсов с указанием ссылки на сервер в сети где крутится Akumuli в ручном режиме. При переходе к (мало-) серийному производству беспилотных автомобилей следует стандартизировать конфигурации и настройки на различных автомбилях. Занятость диска для базы данных Akumuli фиксирована на 32 ГБ с последующей автоматической перезаписью (~5% дискового пространства).

Time Series Data Base (TSDB) Akumuli для систем мониторинга в ЦОД встречалось и раньше (см. на пример [12]). Akumuli - это российская, полностью отечественная разработка, её автор - Евгений Лазин [13].

Ограничения Akumuli, такие как: работа на одиночном компьютере (не поддерживает распределенные системы) и строго хронологический характер данных во временном ряду, снимаются работой Plog-net daemon, передающего данные в базу данных строго хронологически с меткой локального времени и только на локальный экземпляр Akumuli.

Все сервера в кластере на автомобиле поддерживают протокол синхронизации времени, так что расхождения во временных метках на разных узлах является признаком неправильного функционирования кластера и поводом для остановки автомобиля.

Все обращения к базе данных Akumuli записывают идентификатор, включающий имя узла и другую статическую информацию, позволяющую осуществить поиск и выделение тех данных которые имеют отношение к отдельным узлам. Таким образом, каждый узел несет идентичную информацию доступную для полного анализа всех узлов, входящих в автомобильный вычислительный кластер.

Результаты реализации

Анализ активности мониторинга показывает загрузку сети на весь кластер около 4 кБ/с (или 13 МБ/час) – это скорость роста аккумулируемой информации. Таким образом, ресурса Akumuli в 32 ГБ хватит на 100 дней непрерывной работы. Затем база данных затирает наиболее старые данные и не требует корректировки в ручном режиме.

Графический анализ Grafana показывает типичную презентацию, Рис. 2. Выведена верхне-уровневая панель собранных значений для одного узла, где показано (слева-направо и сверху-вниз) значение общего потребления каждого узла, температура охлаждающей жидкости, температура ГПУ, ряд датчиков температуры воздуха и потребление ГПУ. Важно подчеркнуть, что наполнение панелей графической презентации еще не до конца оформлено и меняется по мере опыта работы с системой.

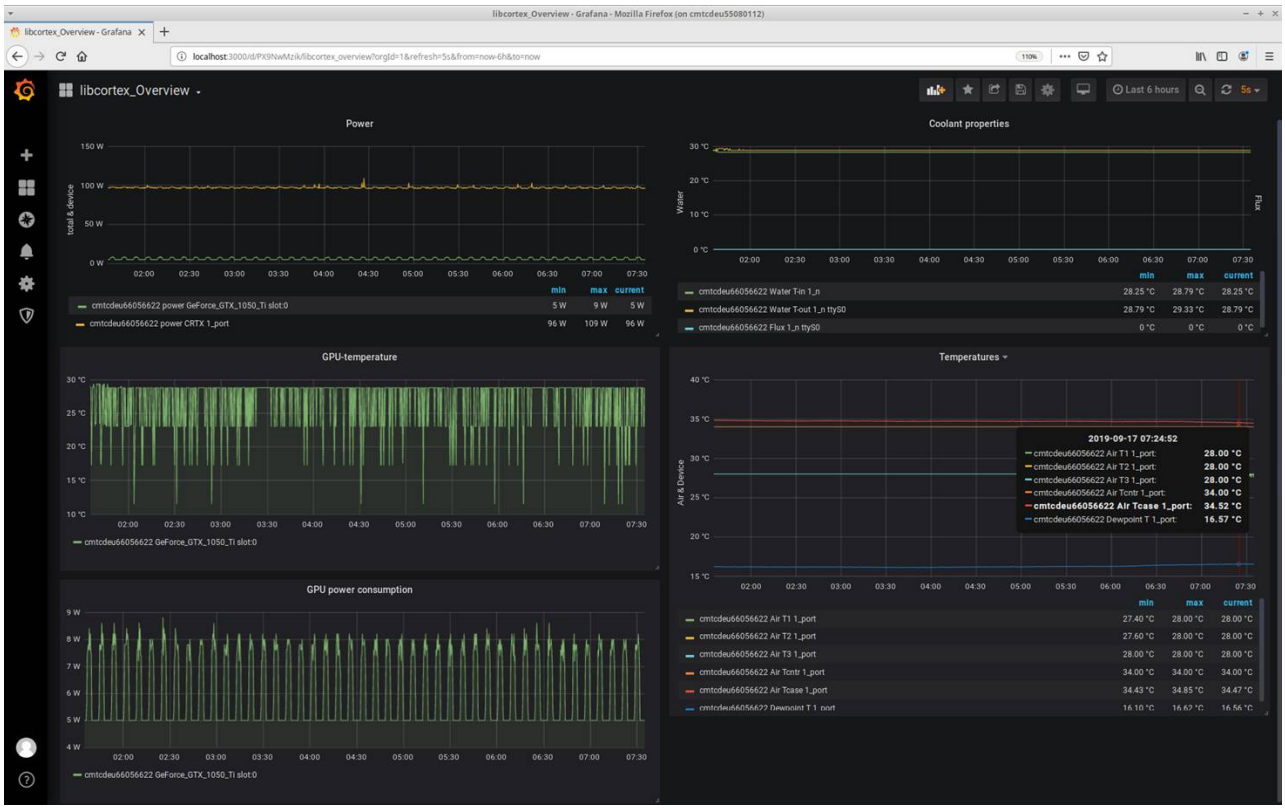

Рис. 2. Типичная презентация информации по граф. ресурсу Grafana в иллюстраторе.

Такая компоновка дана только для иллюстрации. Любая другая собираемая информация может быть представлена для анализа.

Заключение

В работе обсуждается методы и подходы реализации системы мониторинга для высокопроизводительных кластеров в беспилотных

автомобилях. Показаны отличия от программного обеспечения для мониторинга в центрах обработки данных. Проект успешно реализован для проекта тестирования беспилотных автомобилей.

Дальнейшее развитие проекта идет в сторону охвата системой мониторинга самой системы мониторинга, снятия информации с датчиков дополнительных устройств автомобиля и мониторинга программ приложений искусственного интеллекта, управляющих автомобилем.

References

1. SAE J3016 “Levels of Driving Automation”, https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic (дата обращения: 12/05/2019).
2. Nagios. URL: https://www.nagios.org (дата обращения: 12/05/2019).
3. Zabbix. URL: https://www.zabbix.com (дата обращения: 12/05/2019).
4. Most Comprehensive List of Linux Monitoring Tools For SysAdmin, URL: https://www.ubuntupit.com/most-comprehensive-list-of-linux-monitoring-tools-for-sysadmin (дата обращения: 12/05/2019).
5. Top FREE Server Monitoring Tools, URL: https://www.dnsstuff.com/free-server-monitoring-tools (дата обращения: 12/05/2019)
6. In-Time Aviation Safety Management: Challenges and Research for an Evolving Aviation System (2018). ISBN 978-0-309-46880-0, DOI 10.17226/24962
7. Aircracft Health Monitoring – FAA Perspective, presented to IATA Conference by Tim Shaver, 13 November 2017. URL: https://www.iata.org/whatwedo/workgroups/Documents/Paperless_Conference_2017/Day1/1130-1200_AircraftHealthMonitoring_FAA.pdf , (дата обращения: 12/05/2019).
8. NIST Intelligent Systems Division, URL: https://www.nist.gov/el/intelligent-systems-division-73500/ieee-1588, (дата обращения: 12/05/2019).
9. NCURSES Programming HOWTO, URL: https://www.tldp.org/HOWTO/NCURSES-Programming-HOWTO/intro.html, (дата обращения: 12/05/2019).
10. Akumuli. URL: https://akumuli.org/ (дата обращения: 12/05/2019).
11. The open platform for beautiful analytics and monitoring. URL: https://grafana.com (дата обращения: 12/05/2019).
12. Живчикова Н.С., Шевчук Ю.В. Подсистема архивации данных системы мониторинга Botikmon3 // Научный сервис в сети Интернет: труды XX Всероссийской научной конференции (17-22 сентября 2018 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2018. — С. 223-229. URL: http://keldysh.ru/abrau/2018/theses/26.pdf doi:10.20948/abrau-2018-26

13.Лазин Е. Numeric B+tree reference. URL: https://docs.google.com/document/d/1jFK8E3CZSqR5IPsMGojm2LknkNyUZA7tY51N6IgzW_g/pub (дата обращения: 12/05/2019).