

АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РАСПРЕДЕЛЕННОЙ СИСТЕМ ОБРАБОТКИ ДАННЫХ ПРИ АВТОМАТИЗИРОВАННОМ АНАЛИЗЕ ИНФОРМАЦИИ

Устинов Антон Валерьевич,
инженер-программист акционерного общества
«Научно-исследовательский и опытно-
экспериментальный центр интеллектуальных техноло-
гий «Петрокомета»,
г. Санкт-Петербург, Россия,
a.v-ustinov@mail.ru

Охтилев Михаил Юрьевич,
д.т.н., профессор, временный генеральный
директор акционерного общества
«Научноисследовательский и опытно-
экспериментальный центр интеллектуальных
технологий «Петрокомета»,
г. СанктПетербург, Россия,
oxt@mail.ru

АННОТАЦИЯ

В работе рассматривается способ организации одного из этапов обработки измерительной информации, а именно – анализ технического состояния объекта анализа. В качестве объектов анализа рассмотрены: космические аппараты, ракета-носитель и космические системы в целом. Описывается система обработки измерительной информации, модули системы и их функциональные назначения. Способы взаимодействия модулей друг с другом. Приведены требования и ограничения для модулей и системы в целом. Описаны способы решения проблем возникающих на этапах проектирования и реализации систем. Представлено описание архитектура разрабатываемого программного обеспечения распределенной системы обработки данных, относящейся к классу dataflow (архитектура с управлением потоком данных). Распределенная система позволяющая эффективно производить анализ измерительной информации. Описание системы сочетает в себе необходимость ориентации на поток данных, при этом реализуется на существующих аппаратных решениях и может быть легко масштабируема при низки затратах.

Ключевые слова: потоковая модель вычислений; распределенные системы; анализ измерительной информации; мониторинг; dataflow; G-модель; G-сеть.

Для цитирования: Устинов А.В., Охтилев М.Ю. Архитектура программного обеспечения распределенной систем обработки данных при автоматизированном анализе информации // I-methods. 2018. Т. 10. №. 2. С. 31-37.

Введение

Улучшение процесса обработки данных и повышение качества автоматизированного анализа информации, всегда был и остаётся одним из наиболее востребованных направлений в развитии программного обеспечения. Космические средства (КС), космические аппараты (КА), ракета-носитель (РН) и космические системы в целом, на данный момент невозможно представить без системы управления и обработки данных. Во время работы данных объектов, в рамках системы циркулируют огромные потоки информации. Большая часть этой информации, является измерительной.

В рамках данной работы, рассматривается архитектура разрабатываемого программного обеспечения системы обработки данных, относящейся к классу dataflow (архитектура с управлением потоком данных). Система применима для организации одного из этапов обработки измерительной информации (ИИ), а именно – её анализ. Анализ ИИ производится для оценивания технического состояния объектов анализа (ОА) – КА, КС, РН, космических систем в целом и т.д. [1].

Обработка измерительной информации

Информация о состоянии ОА в системе, представляется в виде параметров технического состояния (ТС), среди которых можно выделить измеряемые и вычисляемые. Измеряемыми параметрами ТС, являются значения телеметрируемых параметров ОА. Вычисляемыми параметрами ТС, являются значения вычисляемые по различным алгоритмам на основе значений измеряемых

параметров [2]. В свою очередь, программа вычислений, реализующая алгоритм анализа ИИ и заданная G-сетью [3], представляет из себя, схему программы анализа. Схема программы анализа состоит из операторов, в которых заложены алгоритмы для обработки измеряемых и вычисляемых параметров ТС.

Учитывая специфику ОА и способов сбора информации [4], система, предназначенная для обработки данных, является распределенной на большой территории. В узлах такой системы сосредоточены мощные вычислительные ресурсы, а сами узлы связаны между собой каналами связи.

Обеспечить эффективности обработки ИИ, во многом зависит от организации процесса вычислений в распределенной системе и того, на сколько он отвечает программе вычислений. В настоящее время используются различные средства организации распределенных систем. Очень важным при этом, является процесс их интеграции в уже существующие системы. Зачастую является трудоёмки, дорогостоящим или не возможным без кардинальных изменений текущей системы (в том числе и аппаратных средств). Поэтому, одним из перспективных направлений является подход, который позволяет реализовать высокоэффективные распределенные системы вычислений на уже имеющихся аппаратных средствах. И при необходимости легко масштабировать систему с применением широкого спектра оборудования. Для этого необходимо использовать модульную (сервис-ориентированную) программную систему [5]

Описание системы обработки ИИ

В программу вычислений, заложены принципы потоковой модели вычислений. Принцип потоковой модели – выполнение по готовности данных. Этот же принцип лежит в схеме программы анализа измерительной информации (G – сети) [6] Исходя из этого, в модели вычислений, ключевым моментом является использование локальных данных, прежде всего для хранения и мониторинга набора вычисляемых параметров. Производится это, для нахождения наборов вычисляемых параметров для оператора анализа, а также для временного хранения параметров. Для реализации параллелизма, присущего модели вычислений, в системе необходимо предусмотреть большое количество вычислительных узлов (ВУ). В свою очередь, для обеспечения локальности данных необходимо аналогичное с количеством ВУ, количество модулей, отвечающих за хранение данных. Таким модулем может быть – мастер данных (МД). Все ВУ и МД соединены при помощи коммутационной среды [7].

В идеальном варианте в системе существует автоматическое распределение ресурсов и данных. Конфигурация виртуального вычислительного пространства производится динамически и полностью автоматическим способом [8]. В частности, распределение данных по МД для хранения наборов входных параметров. Но такое предположение справедливо, если коммутационная среда однородна: все времена передач данных между устройствами одинаковы и невелики. Это предположение не реально, на данный момент [9]. Даже в варианте реализации системы на одной вычислительной машине, ее невозможно сделать однородной, так как на работу систему могут влиять внутренние процессы операционной системы и внешние воздействия, а также аппаратные ограничения. Если система функционирует в условия сильной неоднородности коммутационной среды, например, предположить, что система реализована в виде распределенной вычислительной системы, и модули системы размещены на нескольких вычислительных узлах, то результаты могут значительно ухудшиться [10].

Для того чтобы система имела высокие характеристики в условиях неоднородной коммутационной среды, необходимо:

- уметь обнаруживать локальность в схеме программы вычислений, т. е. находить в вычислительном пространстве программы совокупности операторов, внутри которых имеет место очень интенсивное взаимодействие, тогда как между операторами из различных таких совокупностей взаимодействие существенно более слабое;
- построить архитектуру системы, обладающую локальностью данных и операторов, т.е. в структуре системы должны существовать функционально полные программные единицы, включающие в себя наборы операторов и необходимые наборы входных данных с короткими внутрен-

ними связями при высокой пропускной способности этих связей. Между этими программными единицами связи могут быть более длинными, а пропускные способности — меньшими. Из этих программных единиц могут быть образованы более крупные программные единицы, вплоть до получения нескольких уровней иерархии;

- отображать виртуальное пространство вычислений в физическое пространство с сохранением локальности, т. е. тесно связанные между собой вычислительные элементы должны находиться на максимально близких вычислительных машинах [11].

В качестве первого шага к решению проблемы локальности данных и операторов предлагается следующая структура системы:

ВУ и МД объединяются в одну программную единицу — вычислительный модуль (ВМ);

через коммутационную среду передаются только наборы входных параметров с контекстной информацией. Входные параметры, формируемые в МД модуля, передаются на исполнение только в ВУ находящееся в этом модуле;

Предполагается, что все связи внутри ВМ относительно короткие, связи между ВМ — более длинные;

Структура коммутационной среды может быть неоднородной (то есть связи внутри некоторых групп модулей могут быть короче, чем между ВМ из разных групп) [12].

Для исключения нерационального распределения данных в системе, важную роль, играет специалист, проектирующий схему программы анализа. Специалист формирует отдельные задачи, которые будут использоваться в его программе без учёта распределения. Но каждая, такая задача может быть распределена по модулям системы. Таким образом, программа состоит из двух относительно независимых частей: описания алгоритма, не учитывающего распределения вычислительных элементов по модулям и дополнительных указаний, задающих распределение вычислительных элементов по модулям.

Архитектура распределенной системы

Основываясь на приведенных выше данных, можем представить архитектуру распределенной системы, позволяющую эффективно производить анализ ИИ. Описание системы сочетает в себе необходимость ориентации на поток данных, при этом реализуется на существующих аппаратных решениях и может быть легко масштабируема при низки затратах [13].

Среди основных элементов системы, можно выделить следующие:

Коммутационную среда — осуществляет связь элементов системы между собой. Отвечает за передачу команд и данных. В качестве протокола передачи данных предполагается использовать один из открытых протоколов поверх TCP-соединения, например HTTP или WebSocket.;

Вычислительный модуль — модуль (сервис) в системе отвечающий за обработку и хранение данных;

Планировщик вычислений — отвечает за ход выполнения программы вычислений, целостность данных и распределение вычислений и данных между вычислительными модулями.

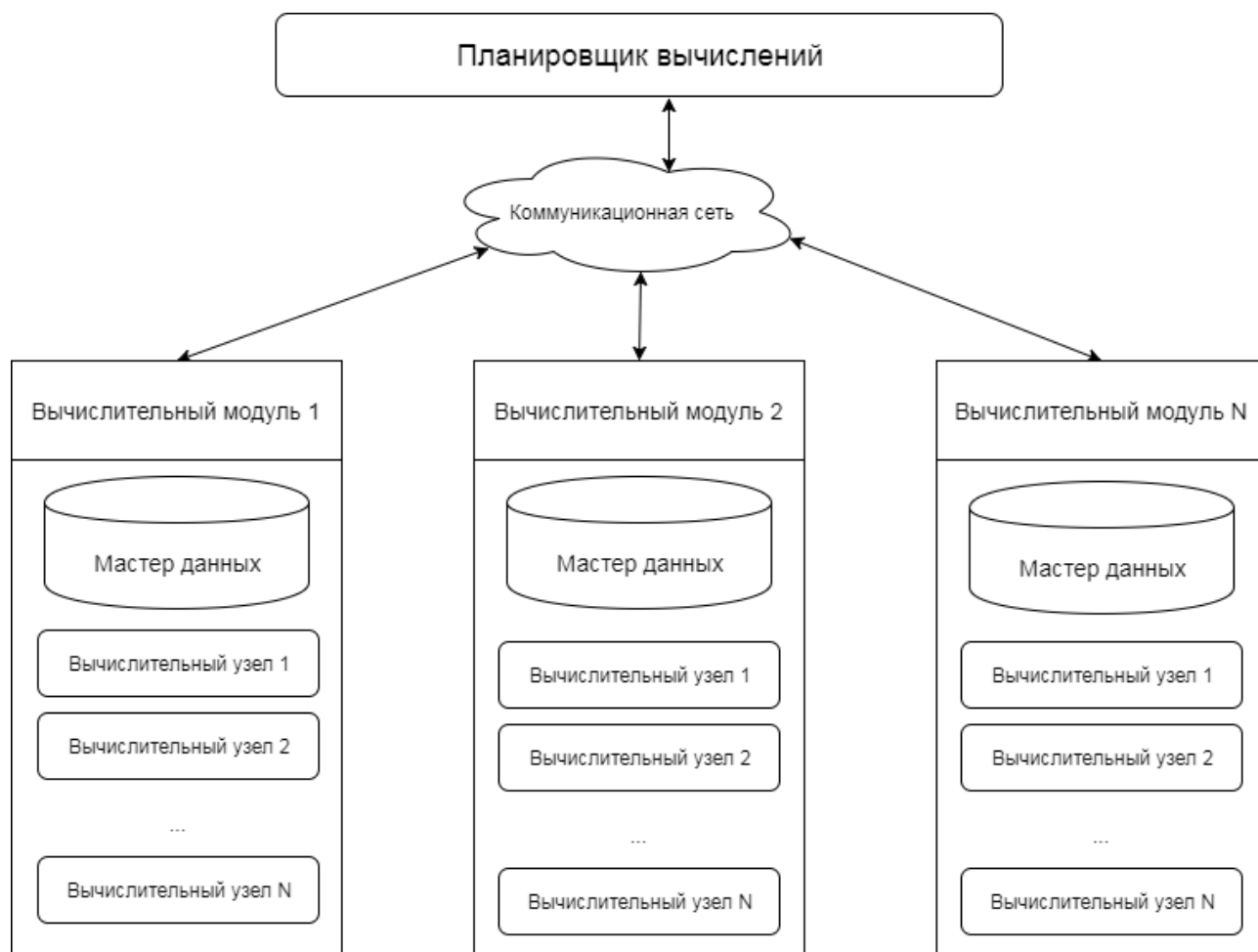


Рис. 1. Архитектура распределенной системы обработки данных

Описание вычислительного модуля

ВМ, в рамках операционной системы можем представить, как приложение (сервис). Одна из фундаментальных основ любого приложения — это процесс или поток. Процесс или поток — это самодостаточный набор инструкций, который операционная система может запланировать для выполнения на ядре процессора. Большинство сложных приложений параллельно запускают множество процессов или потоков по двум причинам:

Чтобы одновременно задействовать больше вычислительных ядер;

Процессы и потоки позволяют проще выполнять параллельные операции.

Процессы и потоки сами по себе расходуют дополнительные ресурсы. Каждый такой процесс или поток потребляет некоторое количество памяти, а кроме того они постоянно подменяют друг друга на процессоре (т. н. переключение контекста). Современные компьютеры могут справляться с сотнями активных процессов и потоков, но производительность сильно страдает, как только заканчивается память или огромное количество операций ввода-вывода приводит к слишком частой смене контекста.

Наиболее простым подходом к построению распределенной системы с большим количеством ВУ — это выделять для каждого ВУ отдельный процесс или поток. Такая архитектура проста для

понимания и легка в реализации, но при этом плохо масштабируется, когда приложению приходится работать с тысячами ВУ одновременно.

Решением в данном случае - использует модель с фиксированным числом процессов для ВУ и очереди операторов для одного вычислительного модуля. Такая модель наиболее эффективно задействует доступные ресурсы системы:

Единственный мастер-процесс выполняет набор основных операций управления системой и обработку очереди операторов, порождает небольшое число рабочих процессов, которые выполняют роль ВУ. Число рабочих процессов в этом случае, не должно превышать количество ядер процессора, что позволит использовать системные ресурсы максимально эффективно. Каждый из ВУ получает оператор, от мастер-процесса и обрабатывает данные в не блокирующемся режиме, минимизируя количество переключений контекста. Каждый рабочий процесс однопоточен и работает независимо, принимая новые операторы и обеспечивая их выполнение.

Описание планировщика вычислений

С точки зрения функциональности планировщик вычислений, не отличается от планировщиков вычислений в других распределенных системах, за исключением того факта, что он ориентирован на выполнение по готовности данных. В работе предложена общая архитектура данного элемента системы.

Процесс планирования в системе может быть представлен следующими этапами:

- отслеживание хода выполнения вычислительной программы;
- отслеживание готовности данных;
- отслеживание ресурсов (доступных VM).

На рис. 1 представлена модель системы планирования в распределенных вычислительных средах, в которой функциональные компоненты связывают два типа потоков данных: прерывистая линия определяет поток ресурсов/поток информации о приложении, прямая линия — поток заданий/поток команд планирования заданий.

Основная работа планировщика заключается в том, что он следит за ходом выполнения вычислительной программы и производит мониторинг готовности данных на VM. При необходимости запуска новых задач, производит поиск подходящих ресурсов в соответствии с полученными от VM информации. В отличие от планировщиков традиционных параллельных и распределенных систем, планировщик не контролирует ресурсы напрямую, а работает как брокер. На рис. 1 показан только один планировщик, однако в действительности может быть развернуто несколько подобных планировщиков для формирования различных структур системы (централизованной, иерархической и децентрализованной) для решения проблем производительности и масштабируемости.

Заключение

Представленная архитектура распределенной системы обработки данных демонстрирует хорошие перспективы для задач, автоматизированного анализа информации. Что может оказать большое влияние на процесс оценивания технического состояния КА, КС, РН, космических систем в целом.

Реализация системы, является не простой и требует дальнейшей работы и исследований, из-за ряда сложностей, возникающих при реализации распределенных систем, основанных на потоковой модели вычислений. В данный момент система находится на стадии разработки. Результат реализации системы будет описан в будущих работах

Литература

1. *Охтилев М.Ю.* Основы теории автоматизированного анализа измерительной информации в реальном времени. Синтез системы анализа. СПб.: ВИКУ им. А.Ф. Можайского, 1999. 161 с.
2. *Охтилев М.Ю., Соколов Б.В., Юсупов Р.М.* Интеллектуальные технологии мониторинга состояния и управления структурной динамикой сложных технических объектов. М.: Наука, 2006. 410 с.
3. *Устинов, А.В. Зянчурин А.Э., Кириллов И.С.* Перспективы использования потоковой модели распределенных вычислений в системах мониторинга // Проблемы управления и моделирования в сложных системах: Труды XIX Международной конференции (Самара 12-15 сентября 2017 г.). Самара, 2017. С. 514-518.

4. Устинов А.В. Применение модели mapreduce для организации распределенных систем мониторинга состояния и управления сложными организационно-техническими объектами // Научная сессия ГУАП. СПб. 2016. Ч. 2. С. 270 – 272.
5. Устинов А.В. Повышение эффективности обработки измерительной информации при проведении пусков ракет-носителей // Материалы V Всероссийской научно-технической конференции «Актуальные проблемы ракетно-космической техники» (Самара, 11-15 сентября 2017 г.). Самара, 2017. С. 251-255.
6. Таненбаум Э., ван Стеен М. Распределенные системы. Принципы и парадигмы: пер. с англ. СПб: Питер, 2003. 877 с.
7. Климов А.В., Левченко Н.Н., Окунев А.С., Змеев Д.Н. Dataflow – парадигма программирования будущего // Тезисы доклада, семинар «Посткремниевые вычисления» (Переславль-Залесский, 24 ноября 2014 г.). Переславль-Залесский, 2014.
8. Климов А.В., Окунев А.С. Поточковая модель вычислений как путь к экзафлопсу // Труды Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: экзафлопсное будущее» (Новороссийск, 19-24 сентября 2011 г.). Новороссийск, 2011. С. 261-266.
9. Климов А.В., Окунев А.С., Степанов А.М. Проблемы развития модели вычислений с управлением потоком данных и особенности ее архитектурной реализации // Материалы международной конференции IT+S&E 07 «Информационные технологии в науке, образовании, телекоммуникации и бизнесе», (Ялта-Гурзуф, май 2007 г.). Ялта-Гурзуф, 2007.
10. Нариньяни А.С. Модель или алгоритм: новая парадигма информационной технологии // Информационные технологии. 1997. № 4. С. 11-16.
11. Тьюгу Э.Х. Концептуальное программирование. М.: Наука, 1984. 255 с.
12. Зубова Т.Н., Мельников Б.Ф. Использование сетей Петри для моделирования процесса принятия управленческих решений // Вектор науки ТГУ. 2011. № 3(17). С. 33-37.
13. Майданович О.В., Охтилев М.Ю., Каргин В.А., Мышко В.В., Соколов Б.В. Теория и практика построения автоматизированных систем мониторинга технического состояния космических средств. СПб.: ВКА, 2011. 219 с.

SOFTWARE ARCHITECTURE FOR DISTRIBUTED DATA PROCESSING SYSTEMS IN AUTOMATED INFORMATION ANALYSIS

Anton V. Ustinov,
Saint-Petersburg, Russia,
a.v-ustinov@mail.ru

Michael Yu. Okhtilev,
Saint-Petersburg, Russia,
oxt@mail.ru

ABSTRACT

Work deals with the organization processing of measuring information is considered, namely the analysis of the technical state and object of analysis. As objects of analysis, space vehicles, a booster rocket and space systems in general are considered. The system processing of measuring information, modules of the system and their functional purposes is described. Modes of interaction of modules with each other. The requirements and limitations for modules and the system as a whole are given. The ways of solving problems arising at the design and implementation stages of systems are described. A description is given of the architecture of the software developed for a distributed data processing system related to the data-flow class. The distributed system allows to effectively analyze the measuring information. The description of the system combines the need to focus on the data stream, while it is implemented on existing hardware solutions and can be easily scaled at a low cost.

Keywords: The streaming model of computation, distributed systems, analysis of measurement data, monitoring, dataflow, G-model, G-network.

References

1. Okhtilev M.Yu. *Osnovy teorii avtomatizirovannogo analiza izmeritel'noj informacii v real'nom vremeni. Sintez sistemy analiza* [Basics of the theory of automated analysis of measurement information in real time. Synthesis of the analysis system]. Saint-Petersburg: VIKU them A.F. Mozhaysky, 1999. 161 p. (In Russian)

2. Okhtilev M.Yu., Sokolov B.V., Yusupov R.M. *Intellektual'nye tehnologii monitoringa sostoyaniya i upravleniya strukturnoj dinamikoj slozhnyh tehniceskix ob'ektov* [Intellectual technologies for monitoring the state and managing the structural dynamics of complex technical objects]. Moscow: Nauka, 2006. 410 p. (In Russian)
3. Ustinov A.V. *Perspektivy ispol'zovaniya potokovoj modeli raspredelennyh vychislenij v sistemah monitoringa* [Prospects of using the flow model of distributed calculations in monitoring systems]. *Problemy upravleniya i modelirovaniya v slozhnyh sistemah: Trudy XIX Mezhdunarodnoj konferencii (Samara 12-15 sentyabrya 2017 g.)* [Complex systems: Control and modeling problems: Proceedings of the XIX International Conference. (Samara, September 12-15, 2017)]. Samara, 2017. Pp. 410-415. (In Russian)
4. Ustinov A.V. *Primenenie modeli mapreduce dlya organizacii raspredelennyh sistem monitoringa sostoyaniya i upravleniya slozhnymi organizacionno-tehniceskimi ob'ektami* [Mapreduce model used in creating distributed systems of monitoring a state and control of complex organizational-technical objects]. *Scientific session of SUAI*. St. Petersburg, 2016. Pt. 2. Pp. 270 – 272. (In Russian)
5. Ustinov A.V. *Povyshenie `effektivnosti obrabotki izmeritel'noj informacii pri provedenii puskov raket-nositelej* [Improvement of efficiency of measurement information processing launch vehicles. Materials of the V All-Russian Scientific and Technical Conference]. *Materialy V Vserossijskoj nauchno-tehniceskoi konferencii «Aktual'nye problemy raketno-kosmicheskoi tehniki» (Samara, 11-15 sentyabrya 2017 g.)* ["Actual problems of rocket and space technology" V Kozlov Readings Samara, September 11-15, 2017]. Samara, 2017. Pp. 251-255. (In Russian)
6. Tanenbaum E., Van Steen M. *Raspredelenny'e sistemy`. Principy` i paradigmy`*. [Distributed systems. Principles and paradigms]. Saint- Petersburg: Peter, 2003. 877 p. (In Russian)
7. Klimov A.V., Levchenko N.N., Okunev A.S., Zmeev D.N. *Dataflow – paradigma programmirovaniya budushhego* [Dataflow - the paradigm of programming the future] *Tezisy` doklada, seminar "Postkremnievy'e vy`chisleniya"* [Abstracts of the report, seminar "Post-silicon calculations", November 24, 2014]. Pereslavl-Zalessky, 2014. (In Russian)
8. Klimov A.V., Okunev A.S. *Potokovaya model' vychislenij kak put' k `ekzaflopsu* [The flow model of computation as a path to the exaflops]. *Trudy Mezhdunarodnoj superkomp'yuternoj konferencii «Nauchnyj servis v seti Internet: `ekzaflopsnoe buduschee» (Novorossiysk, 19-24 sentyabrya 2011 g.)*. [Proceedings of the International Supercomputer Conference "Scientific Service in the Internet: the Exaflop Future" (Novorossiysk, September 19-24, 2011)]. Novorossiysk, 2011. Pp. 261-266. (In Russian)
9. Klimov A.V., Okunev A.S., Stepanov A.M. *Problemy` razvitiya modeli vy`chislenij s upravleniem potokom dannyx i osobennosti ee arhitekturnoj realizacii* [Problems in the development of a computational model with flow control and features of its architectural implementation]. *Materialy mezhdunarodnoj konferencii IT+S&E`07 «Informacionnye tehnologii v nauke, obrazovanii, telekommunikacii i biznese», (Yalta-Gurzuf, maj 2007 g.)*. Yalta-Gurzuf, 2007. ["Information technologies in science, education, telecommunications and business", materials of the international conference IT + S & E`07, Yalta-Gurzuf, May 2007]. Yalta-Gurzuf, 2017. (In Russian)
10. Narinyani A.S. *Model` ili algoritm: novaya paradigma informacionnoj tehnologii* [Model or algorithm: a new paradigm of information technology]. *Informacionny'e tehnologii* [Information technology]. 1997. No. 4. Pp. 11-16. (In Russian)
11. Tyugu E.H. *Konceptual'noe programmirovaniye* [Conceptual programming]. Moscow: Nauka, 1984. 255 p.
12. Zubova T.N., Mel'nikov B.F. *Application of petri nets in modeling managerial decision-making processes*. *Science Vector of Togliatti State University*. 2011. No. 3(17). Pp. 33-37. (In Russian)
13. Majdanovich O.V., Okhtilev M.Ju., Kargin.V.A., Myshko.V., Sokolov B.V. *Teoriya i praktika postroeniya avtomatizirovannyx sistem monitoringa texniceskogo sostoyaniya kosmicheskix sredstv* [Theory and practice of constructing automated systems for monitoring the technical state of space vehicles]. Saint-Petersburg: VIKU them. Mozhaisky, 2011. (In Russian)

Information about authors:

Ustinov A.V, Postgraduate student, software developer of Scientific Research and Experimental Center of Intelligent Technologies “Petrokometa”;

Okhtilev M.Yu., PhD, Full Professor, temporary CEO of Scientific Research and Experimental Center of Intelligent Technologies “Petrokometa”.

For citation: Ustinov A.V., Okhtilev M.Yu. Software architecture for distributed data processing systems in automated information analysis. *I-methods*. 2018. Vol. 10. No. 2. Pp. 31-37. (In Russian)