

Информационные технологии

Аппаратные ускорители для СУБД на основе GPU и DPU

А.В. Соловьев

Федеральный исследовательский центр «Информатика и управление»
Российской академии наук, г. Москва, Россия

Аннотация. В статье выполнен обзор применения аппаратных ускорителей для СУБД на основе GPU и DPU. Сделан обзор подобных аппаратных ускорителей, рассмотрены практические примеры их использования. Определены достоинства и недостатки применения GPU и DPU. На основе проведенного обзора сделан вывод, что подход позволяет ускорять операции в БД, связанные с обработкой запросов, операциями ввода и вывода, распараллеливания обработки данных БД. Однако применение таких ускорителей усложняет систему в целом, снижает ее энергоэффективность.

Ключевые слова: системы управления базами данных, СУБД, GPU, DPU, аппаратные ускорители обработки данных.

DOI: 10.14357/20790279250405 **EDN:** PGXZUD

Введение

Критичность скорости выполнения операций обработки данных в условиях цифровой экономики становится все актуальнее. Это касается и отдельных информационных систем масштаба предприятия, и, тем более, отрасли или экономики в целом. Требования к высокой согласованности данных на узлах территориально-распределенной СУБД также становятся все более критичными, иными словами, все последние изменения данных всегда должны быть доступны пользователям в режиме онлайн.

Понимая это, производители СУБД пытаются повысить производительность своих продуктов не только за счет совершенствования алгоритмов поиска, применения методов искусственного интеллекта (ИИ), но и с помощью различных аппаратных ускорителей, на которых, кстати, также могут быть реализованы алгоритмы ИИ. К таким аппаратным ускорителям относят GPU (Graphics Processing Unit (графический процессор)) и DPU

(Data Processing Unit (сопроцессор для обработки данных)), ставшими популярными за счет возможности разгрузить CPU (Central Processing Unit (центральный процессор)) от целого ряда операций обработки данных.

Статья имеет целью выполнить обзор применения аппаратных ускорителей СУБД с помощью GPU и DPU. В обзоре выполнен анализ примеров применения GPU и DPU в СУБД, выявлены достоинства и недостатки их применения.

1. Краткий обзор применения GPU для ускорения операций в СУБД

Для начала, нужно сказать, что GPU – это программируемое устройство (графический процессор), предназначенное для быстрой визуализации изображений и видео с высоким разрешением. Особенностью GPU является то, что они могут обрабатывать множество фрагментов данных од-

новременно, т.к. предназначены для быстрой визуализации изображений и видео с высоким разрешением. Способность к параллельной обработке массивов данных и послужила толчком к использованию GPU в СУБД.

В настоящее время на рынке появился целый ряд решений, который использует GPU в качестве аппаратных ускорителей. Кратко опишем их возможности.

Безусловно, промышленным примером использования аппаратных ускорителей на основе GPU можно назвать продукт компании Alibaba Cloud (КНР) – Elastic GPU Service (EGS). Это вычислительный сервис на базе GPU, который разработан для реализации алгоритмов глубокого обучения, обработки видео, научных вычислений и визуализации. EGS использует следующие графические процессоры: AMD FirePro S7150, NVIDIA Tesla M40, NVIDIA Tesla P100, NVIDIA Tesla P4, NVIDIA Tesla V100 (см. подробнее [1]). О конкретных примерах внедрения данного продукта судить достаточно сложно, но известно, что в настоящее время 40% из 500 крупнейших предприятий Китая используют продукты компании Alibaba Cloud, а также около половины китайских компаний, зарегистрированных на бирже (см., например, [2]).

Еще одним примером использования GPU именно в СУБД является СУБД Kinetica. Архитектура данной СВУБД адаптирована для работы в кластере из множества узлов СУБД без разделяемых ресурсов. Для хранения данных используется столбцовый формат. Для хранения данных на различных узлах БД используются различные стратегии сегментирования между узлами. По данным открытой печати данную СУБД использует Почтовая служба США. Из конфигурации использования СУБД известно, что БД распределена на 200 узлов БД, может обрабатывать до 150 млрд записей БД/с, использует платформу Power9 [3]. Использование платформы Power9, по-видимому, обусловлено наличием в процессоре Power9 шины NVLink, напрямую соединяющие GPU на скорости до 200 Гб/с и позволяющей нескольким GPU обмениваться данными без их пересылки в оперативную память. Это дает возможность фактически объединить несколько GPU в рамках узла БД для обработки больших массивов данных.

Еще одной СУБД, реализованной на платформе Power9 и использующей аппаратные ускорители GPU, является облачная графовая СУБД Amazon Neptune (113 место в рейтинге DB-Engine по количеству установок). С помощью GPU ускоряются операции параллельной обработки данных на графах СУБД.

Также на платформе Power9 реализована еще одна СУБД, использующая GPU – ScreamDB, впрочем, имеющая достаточно низкий рейтинг по количеству установок (234 место в рейтинге DB-Engine).

Следующим примером является отечественная BI-система «Полиматика». Она обладает системой хранения данных на основе столбцовового формата. Интересным фактом является то, что для выполнения операций сортировки и кластеризации данных используются GPU [3].

Следующие примеры показывают, что GPU можно использовать для ускорения запросов в СВУБД. Так, например, относительно недавно былложен в открытый доступ код SQL-движка BlazingSQL, использующего GPU для ускорения запросов [4]. Хотя BlazingSQL не является СУБД, но позиционируется как движок для анализа и обработки больших наборов данных, сравнимый по своим задачам с Apache Spark. Код BlazingSQL написан на языке Python и открыт под лицензией Apache 2.0. Благодаря распараллеливанию операций в GPU, выполнение запросов осуществляется до 20 раз быстрее, чем в Apache Spark. Для работы с GPU применяется развивающийся при участии компании NVIDIA набор открытых библиотек RAPIDS, позволяющий создавать приложения для обработки данных и аналитики, выполняемые целиком на стороне GPU.

Другими примерами использования GPU для ускорения запросов в СУБД можно считать PG-Strom, позволяющий СУБД PostgreSQL работать быстрее при использовании GPU [5], а также новый процедурный язык для СУБД PostgreSQL – PgOpenCL, также позволяющий использовать GPU, а также GPGPU (General Purpose GPU (non-graphics programming on a GPU)) для ускорения выполнения запросов к данным [6].

Следующее экспериментальное исследование эффективности применения GPU для облачных решений [7] показывают, что производительность информационной системы можно повысить за счет увеличения общего времени использования GPU. При обработке потоков данных такое ускорение приводит к увеличению параллелизма обработки на уровне приложения.

В качестве вывода из краткого обзора можно сделать заключение, что GPU может использоваться для ускорения операций обработки данных СУБД. В первую очередь такой аппаратный ускоритель позволяет высвободить ресурсы CPU, а также распараллелить обработку запросов.

К преимуществам GPU можно отнести:

- возможность значительного распараллеливания обработки данных, в том числе их записи, а также для выполнения и оптимизации запросов;

- высокая скорость обработки данных.
- Среди недостатков GPU можно выделить:
- проблема снижения скорости передачи данных между CPU и GPU в случае использования общей памяти, в том числе виртуальной и долговременной;
- сложности построения гетерогенных планов запросов из-за необходимости синхронизации быстродействия в связке CPU-GPU;
- низкая энергоэффективность системы из-за высокого уровня энергопотребления GPU.

Кроме недостатков существуют проблемы использования GPU:

- первая проблема лежит не совсем в технической сфере – это зависимость от поставок оборудования GPU, которая может оказаться недоступной из-за сложной международной установки. Отечественных производителей GPU не существует;
- не каждый алгоритм подходит для использования GPU с выигрышем в производительности. Исследования показывают, что алгоритмы для GPU должны быть адаптированы для параллельной обработки в стиле один поток команд, много потоков данных (см., например, [8, 9]);
- несмотря на множество академических исследований, готовых к выпуску на рынок СУБД на базе GPU крайне мало, а их рейтинг по количеству установок невысок. В качестве дополнительного аргумента в пользу этой проблемы говорит тот факт, что крупные производители СУБД не торопятся адаптировать свои разработки под GPU. Так, например, согласно заявлениям крупнейшего производителя СУБД Oracle (1 место в рейтинге DB-Engine по количеству установок), «Oracle активно работает над адаптацией своих алгоритмов работы с БД для использования преимуществ GPU и выпустит эти алгоритмы, если прирост производительности достаточен и устойчив» (см., например, [10]).

Несмотря на проблемы и недостатки, в целом можно отметить, что использование GPU представляется перспективным для будущего развития СУБД в плане ускорения обработки данных. Предположительно использование для распараллеливания обработки данных, в том числе их записи, оптимизации запросов может существенно повысить эффективность работы СУБД. Однако существуют и альтернативные подходы к ускорению, такие как гибридные планы выполнения запросов, использование методов ИИ. Этим исследованиям будут посвящены дальнейшие работы автора статьи.

2. Краткий обзор применения DPU для ускорения операций в СУБД

DPU — это специальные процессоры для обработки данных, представляют собой разновидность умных сетевых карт. Главная цель использования DPU в СУБД — разгрузить CPU, приняв на себя операции по вводу и выводу данных.

Необходимо отметить, что в отличие от GPU, пока примеров промышленного использования DPU в СУБД не существует. Однако многие крупные разработчики информационных систем, облачных решений и СУБД используют DPU в dataцентрах.

Приведем ряд примеров использования DPU. Google производит тензорные процессоры Google TPU ASIC (Application-Specific Integrated Circuit). Это специализированный процессор для реализации алгоритмов машинного обучения. Он используется для запуска передовых моделей машинного обучения с помощью сервисов ИИ в Google Cloud [11,12].

В data-центрах Amazon используются DPU AWS Graviton на ядре 64-битных ARM общего назначения. AWS Graviton используются для оптимизации соотношения цены и производительности в рабочих нагрузках, требующих больших вычислительных ресурсов [11,12].

Исследовательский проект DAG-процессор (DPU-v2) позволил создать специализированный процессор для эффективной обработки потоков данных в вероятностном искусственном интеллекте. Архитектура процессора включает древовидный путь передачи данных для эффективного повторного использования данных, специализированный банк регистров и целевых межсоединений, настроенных для поддержки нерегулярного доступа к регистрам. DPU-v2 эффективно используется с помощью специального компилятора, который систематически сопоставляет операции путем передачи данных, минимизирует конфликты между банками регистров и позволяет избежать проблемы с конвейером. Исследование пространства проектирования позволяет определить оптимальную конфигурацию архитектуры, которая минимизирует энергозатраты. Экспериментальные данные показывают увеличение производительности от 1,4 до 14 раз в операциях по обработке данных [13,14].

Компания NVIDIA разрабатывает архитектуру высокопроизводительной вычислительной системы под названием Cloud Native Supercomputing. В архитектуру включен новый компонент – DPU, который представляет собой систему на кристалле (SoC), включающую сетевой адаптер InfiniBand (IB) и Ethernet, программируемые ядра ARM, память, коммутаторы PCI и специализированные

ускорители. Устройства BlueField-1 и BlueField-2 являются первыми экземплярами DPU от NVIDIA. DPU используются для ускорения управления файловой системой, мониторинга и разгрузки поддержки коммуникационных библиотек [15].

Для ускорения сканирования больших таблиц графовой СУБД Poseidon Graph разработана архитектура UPMEM. Архитектура основана на подходе «Обработка в памяти» (PIM – Processing In Memory). Ядром архитектуры UPMEM является модуль DIMM (Dual In-line Memory Module) UPMEM, который основан на обычном модуле DIMM DDR4-2400, но оснащен дополнительными микросхемами PIM. Модули DIMM UPMEM организованы в ряды. Модуль DIMM UPMEM состоит из двух рядов, каждый из которых состоит из 8 микросхем с поддержкой PIM. Микросхема PIM обычно состоит из 8 блоков обработки DRAM (DPU). Каждый DPU имеет эксклюзивный доступ к 64 МБ основной оперативной памяти (MRAM), 24 КБ оперативной памяти для команд (IRAM) и 64 КБ рабочей оперативной памяти (WRAM) для обработки данных. Поскольку DPU имеют доступ только к своей собственной MRAM, между различными DPU невозможна прямая связь. Кроме того, DPU состоит из 32-битного ядра RISC общего назначения с максимальной достижимой частотой 400 МГц, которое может выполнять специальный набор команд в многопоточном конвейере. Для многопоточности доступно 24 аппаратных потока. Все потоки используют одну и ту же память DPU, что требует синхронизации для обеспечения согласованности при обновлении общей памяти. Эта архитектура позволяет выполнять одну и ту же программу параллельно с разными фрагментами данных непосредственно в памяти без участия процессора. Каждый DPU может использовать до 24 потоков данных, которые могут выполняться параллельно. Для эксперимента было передано 10 000 фрагментов данных по 817 записей размером 625 МБ с хоста на разное количество DPU (от 16 до 510) и фрагменты были равномерно распределены между всеми DPU. В результате достигнуто существенное ускорение при одновременном снижении энергопотребления за счет использования DPU (см. [16]).

На основании краткого обзора можно утверждать, что преимуществом использования DPU для аппаратного ускорения операций СУБД является возможность значительного ускорения за счет распараллеливания обработки данных, а также ускорения операций ввода/вывода при передаче данных.

К недостаткам использования DPU можно отнести:

- проблема синхронизации производительности при обработке данных в связке с CPU и GPU;
- трудности построения гетерогенных планов запросов CPU-GPU-DPU.

Также можно обозначить проблемы, в целом аналогичные при использовании GPU:

- несмотря на большое количество исследований, крайне мало промышленных образцов;
- такая же, как и для GPU зависимость от поставок оборудования, которое может оказаться недоступным из-за сложной международной установки. Отечественных производителей GPU не существует;
- потенциально те же проблемы передачи данных в связке CPU-DPU, что и у GPU. Необходимы дополнительные исследования потери производительности при передаче данных в связке CPU-DPU-GPU.

Оценки производительности современных систем показывают, что до 30% нагрузки на CPU идет за счет функций ввода вывода данных (см., например, [17]). В этой связи использование DPU представляется перспективным для перемещения данных и их обработки, в том числе, при реплицировании и партиционировании данных БД, несмотря на заявленные недостатки и проблемы.

Заключение

В статье выполнен обзор аппаратных ускорителей операций СУБД с помощью GPU и DPU. Рассмотрены возможностей GPU и DPU, а также практические примеры их использования, в том числе и промышленные.

Определены достоинства и недостатки применения GPU и DPU. Отмечен ряд проблем их применения и альтернативные технологии ускорения обработки данных, которым будут посвящены будущие исследования автора.

В качестве вывода по применению GPU и DPU для ускорения обработки данных в СУБД, можно утверждать, что подход позволяет ускорять операции в БД, связанные с обработкой запросов, параллельной обработки данных.

Возможно, перспективным является использование следующей архитектуры СУБД для ускорения операций обработки данных за счет разгрузки CPU:

- CPU используется для вычислений общего назначения (~40% операций);
- GPU используется для ускорения параллельной обработки данных при планировании, выполнении, оптимизации запросов к БД (~30% операций);

- DPU используется для операций передачи данных, ввода, вывода и перемещения данных (в том числе партиционирование, репликация, сжатие/шифрование, ~30% операций).

Вместе с тем следует отметить, что применение GPU и DPU усложняет систему в целом и увеличивает общую стоимость владения. Требуют исследований эффективности совместного использования CPU-DPU-GPU от разных производителей.

В дальнейших исследованиях планируется рассмотреть альтернативные технологии ускорения выполнения операций обработки данных в СУБД с помощью методов искусственного интеллекта, гибридных планов выполнения запросов, использования архитектур RISC-V.

Литература

1. Alibaba Cloud Documentztion: Alibaba Cloud // URL: <https://www.alibabacloud.com/help/en?spm=a2c63.l28256.0.0.2ec01ef3Z8hJVf> (дата обращения 29.04.2025).
2. Wu G., Chen Z., Dang J. Big Data Enabled Computing // Intelligent Bridge Maintenance and Management. Springer Tracts in Civil Engineering. 2024. Springer, Singapore. https://doi.org/10.1007/978-981-97-3827-4_5.
3. Волков Д., Николаенко А. На пути к «железным» СУБД // Открытые системы. СУБД, №02, 2019 : OSP. URL: <https://www.osp.ru/os/2019/02/13054946> (дата обращения 29.04.2025).
4. Открыт код SQL-движка BlazingSQL, использующего GPU для ускорения : OpenNET // URL: <https://www.opennet.ru/opennews/article.shtml?num=51222> (дата обращения 29.04.2025).
5. Zvonko K. GPU Accelerated SQL queries with PostgreSQL & PG-Strom in OpenShift-3.10 : RedHat // URL: <https://www.redhat.com/en/blog/gpu-accelerated-sql-queries-with-postgresql-pg-strom-in-openshift-3-10> (дата обращения 29.04.2025).
6. Child T. Introducing PgOpenCL A New PostgreSQL Procedural Language Unlocking the Power of the GPU : GPGPU Accelerated Database. URL: <https://wiki.postgresql.org/images/6/65/Pgopencl.pdf> (дата обращения 29.04.2025).
7. Sánchez-Ribes V., Maciá-Lillo A., Mora H. et al. Efficient GPU Cloud architectures for outsourcing high-performance processing to the Cloud. Int J Adv Manuf Technol. 2024;133:949–958.
8. Wheat J. Implementing SQL on GPUs. Part 1 : sqream. com // URL: <https://sqream.com/blog/implementing-sql-on-gpus-part-1/> (дата обращения 29.04.2025).
9. Wheat J. Implementing SQL on GPUs. Part 2 : sqream.com // URL: <https://sqream.com/implementing-sql-on-gpus-part-2/> (дата обращения 29.04.2025).
10. Colgan M. Does GPU hardware help Database workloads? : Oracle Database Insider // URL: <https://blogs.oracle.com/database/post/does-gpu-hardware-help-database-workloads> (дата обращения 29.04.2025).
11. Как устроены DPU, сопроцессоры для обработки данных. Блог компании Selectel : habr. URL: <https://habr.com/ru/companies/selectel/articles/515446/> (дата обращения 29.04.2025).
12. Что такое DPU и как они устроены? : timeweb. URL: <https://timeweb.cloud/blog/dpu> (дата обращения 29.04.2025).
13. Shah N., Meert W., Verhelst M. DAG Processing Unit Version 2 (DPU-v2): Efficient Execution of Irregular Workloads on a Spatial Datapath // Efficient Execution of Irregular Dataflow Graphs. 2023. Springer, Cham. https://doi.org/10.1007/978-3-031-33136-7_5.
14. Shah N., Meert W., Verhelst M. DAG Processing Unit Version 1 (DPU): Efficient Execution of Irregular Workloads on a Multicore Processor // Efficient Execution of Irregular Dataflow Graphs. 2023. Springer, Cham. https://doi.org/10.1007/978-3-031-33136-7_4.
15. Shainer G. et al. NVIDIA’s Cloud Native Supercomputing // Nichols, J., et al. Driving Scientific and Engineering Discoveries Through the Integration of Experiment, Big Data, and Modeling and Simulation. SMC 2021. Communications in Computer and Information Science. 2022;1512. Springer, Cham. https://doi.org/10.1007/978-3-030-96498-6_20.
16. Baumstark A., Jibril M.A., Sattler KU. Accelerating Large Table Scan Using Processing-In-Memory Technology // Datenbank Spektrum. 2023;23:199–209. <https://doi.org/10.1007/s13222-023-00456-z>.
17. Нагрузка на сервер: определение причин : FirstVDS // URL: <https://firstvds.ru/technology/nagruzka-na-server-opredelenie-prichin> (дата обращения 29.04.2025).

Соловьев Александр Владимирович. Федеральный исследовательский центр «Информатика и управление» Российской академии наук» г. Москва, Россия. Главный научный сотрудник. Доктор технических наук. Область научных интересов: системный анализ, системы управления базами данных, теория надежности, математическое моделирование, долговременное хранение электронных документов. E-mail: soloviev@isa.ru

GPU and DPU hardware accelerators for DBMS

A.V. Solovyev

Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, Moscow, Russia

Abstract. The article provides an overview of the use of GPU and DPU hardware accelerators for DBMS. A review of such hardware accelerators is performed, and practical examples of their use are considered. The advantages and disadvantages of using GPUs and DPUs have been identified. Based on the review, it was concluded that the approach makes it possible to speed up database operations related to query processing, input and output operations, and parallelization of database data processing. However, the use of such accelerators complicates the system as a whole, reducing its energy efficiency. In further research, it is planned to consider other technologies for accelerating the execution of operations in the database.

Keywords: database management systems, DBMS, GPU, DPU, hardware accelerators for data processing

DOI: 10.14357/20790279250405 **EDN:** PGXZUD

References

1. Alibaba Cloud Documentztion: Alibaba Cloud. URL: <https://www.alibabacloud.com/help/en?spm=a2c63.l28256.0.0.2ec01ef3Z8hJVf>. Last access: 29.04.2025.
2. Wu G., Chen Z., Dang J. 2024. Big Data Enabled Computing. Intelligent Bridge Maintenance and Management. Springer Tracts in Civil Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-97-3827-4_5.
3. Volkov D., Nikolayenko A. 2019. On the way to the “iron” DBMS. Open systems. DBMS, №02: OSP. URL: <https://www.osp.ru/os/2019/02/13054946>. Last access: 29.04.2025.
4. The code of the Blazing SQL SQL engine has been opened, which uses the GPU to accelerate : OpenNET. URL: <https://www.opennet.ru/opennews/art.shtml?num=51222>. Last access: 29.04.2025.
5. Zvonko K. 2025. GPU Accelerated SQL queries with PostgreSQL & PG-Strom in OpenShift-3.10 : RedHat // URL: <https://www.redhat.com/en/blog/gpu-accelerated-sql-queries-with-postgresql-pg-strom-in-openshift-3-10>. Last access: 29.04.2025.
6. Child T. 2025. Introducing PgOpenCL A New PostgreSQL Procedural Language Unlocking the Power of the GPU : GPGPU Accelerated Database. URL: <https://wiki.postgresql.org/images/6/65/Pgopencl.pdf>. Last access: 29.04.2025.
7. Sánchez-Ribes V., Maciá-Lillo A., Mora H. et al. Efficient GPU Cloud architectures for outsourcing high-performance processing to the Cloud. Int J Adv Manuf Technol. 2024;133:949–958.
8. Wheat J. 2025. Implementing SQL on GPUs. Part 1 : sqream.com. URL: <https://sqream.com/blog/implementing-sql-on-gpus-part-1/>. Last access: 29.04.2025.
9. Wheat J. 2025. Implementing SQL on GPUs. Part 2 : sqream.com. URL: <https://sqream.com/implementing-sql-on-gpus-part-2/>. Last access: 29.04.2025.
10. Colgan M. 2025 Does GPU hardware help Database workloads? : Oracle Database Insider. URL: <https://blogs.oracle.com/database/post/does-gpu-hardware-help-database-workloads>. Last access: 29.04.2025.
11. How DPUs and coprocessors for data processing are arranged. Selectel company blog : habr. URL: <https://habr.com/ru/companies/selectel/articles/515446/>. Last access: 29.04.2025.
12. What is DPU and how are they arranged? : timeweb. URL: <https://timeweb.cloud/blog/dpu>. Last access: 29.04.2025.
13. Shah N., Meert W., Verhelst M. DAG Processing Unit Version 2 (DPU-v2): Efficient Execution of Irregular Workloads on a Spatial Datapath. Efficient Execution of Irregular Dataflow Graphs. Springer, Cham. 2023. https://doi.org/10.1007/978-3-031-33136-7_5.
14. Shah N., Meert W., Verhelst M. DAG Processing Unit Version 1 (DPU): Efficient Execution of

- Irregular Workloads on a Multicore Processor. Efficient Execution of Irregular Dataflow Graphs. Springer, Cham. 2023. https://doi.org/10.1007/978-3-031-33136-7_4.
15. Shainer G. et al. NVIDIA's Cloud Native Supercomputing. Nichols, J., et al. Driving Scientific and Engineering Discoveries Through the Integration of Experiment, Big Data, and Modeling and Simulation. SMC 2021. Communications in Computer and Information Science. 2022;1512. Springer, Cham. https://doi.org/10.1007/978-3-030-96498-6_20.
16. Baumstark A., Jibril M.A., Sattler KU. Accelerating Large Table Scan Using Processing-In-Memory Technology. Datenbank Spektrum, 2023;23:199–209. 2023. <https://doi.org/10.1007/s13222-023-00456-z>.
17. Server load: determining the causes : FirstVDS // URL: <https://firstvds.ru/technology/nagruzka-na-server-opredelenie-prichin>. Last access: 29.04.2025.

Aleksandr V. Solovyev. Chief Researcher, Doctor of Technical Sciences. Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 44/2 Vavilova str., Moscow, 119333, Russia. E-mail: soloviev@isa.ru