

Эльбрус как архитектура для задач НРС

Е.О. Тютляева¹, И.О. Одинцов², А.А. Московский¹, С.С. Конюхов², Г.В. Мармузов¹,
М.И. Нейман-заде³, А.А. Аникин³, В.Е. Логинов³, В.Ю. Волконский³

¹ЗАО «РСК Технологии»,

²ООО «РСК Лабс»,

³АО «МЦСТ»

Одним из необходимых условий, обеспечивающих высокую скорость разработки программного кода и эффективность выполнения расчётной задачи на суперкомпьютере, является наличие зрелой программной экосистемы, позволяющей запускать большие и сложные высокопроизводительные приложения с множеством программных зависимостей. В данной статье проведён обзор программного обеспечения для архитектуры Эльбрус, предназначенного для проведения высокопроизводительных вычислений.

В статье проанализирован базовый системный инструментарий (компиляторы и библиотеки для эффективного исполнения программ в общей и распределённой памяти), степень готовности математических библиотек, а также рассмотрены подходы к переносу и оптимизации сложных программных пакет и библиотек.

Ключевые слова: экосистема Эльбрус, стек программного обеспечения для Эльбрус, высокопроизводительные библиотеки для Эльбрус, архитектура VLIW, микропроцессоры Эльбрус

1. Введение

Несмотря на то, что существуют узкоспециализированные суперкомпьютеры, созданные для решения специфических задач, большая часть суперкомпьютеров и центров обработки данных должна быть готова к задачам из разных прикладных областей. Для быстрого выполнения конкретных задач на суперкомпьютерах необходимо не только высокопроизводительное вычислительное оборудование, но и наличие оптимизированного как системного, так и прикладного ПО.

Архитектура Эльбрус относится к классу VLIW (Very Long Instruction Word), таким образом отличительной особенностью архитектуры является явный параллелизм исполнения отдельных операций. Технологии оптимизирующей компиляции с аппаратной поддержкой параллелизма позволяют обнаруживать параллелизм элементарных инструкций кода и планировать операции в широкие команды, которые запускаются на исполнение одновременно в одном такте микропроцессора.

Вопросы построения высокопроизводительных вычислительных систем на базе отечественной аппаратно-программной платформы «Эльбрус» уже рассматривались в статьях [1, 2]. В данной работе мы сконцентрируемся на изучении доступного в настоящее время прикладного программного обеспечения и специализированных библиотек, оптимизированного для архитектуры Эльбрус, который необходим для решения широкого класса высокопроизводительных задач. Наличие необходимого прикладного ПО часто является определяющим фактором при выборе архитектуры для построения суперкомпьютера. Очень важно, чтобы основные усилия были направлены на разработку соответствующего программного стека, который будет поддерживать существующие аппаратные технологии.

В данной статье проводится обзор системного программного обеспечения для архитектуры Эльбрус (основной акцент будет сделан на ПО для Эльбрус-8С), необходимого для поддержки параллельного выполнения программ, а также различных специализированных библиотек, которые могут быть использованы для разработки высокопроизводительного прикладного ПО.

2. Базовый системный инструментарий для поддержки параллельных программ

В настоящее время дистрибутив ОС «Эльбрус» включает средства разработки для основных распространенных языков программирования. Заложенная в архитектуру собственная система команд, не имеющая точных аналогов в мире, требует разработки собственного компилятора. Текущим итогом разработки является выпущенная версия 1.23 проприетарного компилятора `lcc` (`Elbrus C Compiler`) в составе системы программирования, совместимой с широко распространенной системой компиляторов `Gnu Compiler Collection` (`gcc`) версии 5.5.0. Текущее состояние средств разработки для языков программирования высокого уровня:

- C. Номинальная совместимость с `gcc-7.3.0`, Стандарты C90, C99 поддержаны полностью, C11 (ISO/IEC 9899:2011) поддержан полностью за исключением необязательного расширения `Atomic`.
- C++. Номинальная совместимость с `g++-7.3.0`. Стандарты C++03, C++11, C++14 поддержаны полностью, для C++17 поддержка экспериментальная.
- Fortran. Номинальная совместимость с `gfortran-5.5.0`, есть неполная совместимость с `gfortran-6.5.0` и `gfortran-7.3.0`. Стандарт Fortran 95 (final draft ISO/IEC 1539-1:1997) поддержан полностью, стандарты Fortran 2003, Fortran 2008 и Fortran 2018 поддержка экспериментальная.
- Java. Номинальная совместимость с Java 8, разработан оптимизирующий JIT-компилятор на базе пары: переносимого интерпретатора Java и компилятора байт-кода Java на основе оптимизатора собственной разработки.
- JS. Успешно используется оптимизирующий JIT-компилятор, реализован транслятор+система поддержки JavaScript – SpiderMonkey (Firefox) и V8 (Chrome).
- C#. При помощи JIT-компилятора реализована система Mono.

Реализация `OpenMP` в оптимизирующем компиляторе `lcc` поддержана на уровне стандарта 3.1.

Реализация стандарта `MPI` предоставляется библиотеками `MPICH-3` и `OpenMPI-2`. Поддерживается работа `OpenMPI-2` с применением наборов библиотек `OFED` (`OpenFabrics Enterprise Distribution`) и `OpenUCX` (`Unified Communication X`) для высокопроизводительных сетей с функциями `RDMA` и `kernel bypass`. Примером таких сетей является `Infiniband`, а также адаптеры `Ethernet` с поддержкой технологий `RoCE` (`RDMA over Converged Ethernet`) или `iWarp` (`internet Wide Area RDMA Protocol`).

3. Обзор основных функций математических (вычислительных) библиотек

3.1. Математические функции

Одним из основных требований экосистемы НРС является наличие набора оптимизированных, высокопроизводительных математических библиотек. Для архитектуры Эльбрус разработана высокопроизводительная математическая и мультимедийная библиотека `EML` (`Elbrus Mathematical and Multimedia Library`) [3]. Ниже приведена краткая классификация реализованных математических функций, входящих в библиотеку.

- Ядро (`Core`). Базовые функции обращения к памяти (заказ и освобождение памяти), информация о библиотеке (номер версии, статус).

- Вектор (Vector). Базовые операции с векторами, векторные математические функции, статистика, преобразования, нормирование и т.п.
- Линейная Алгебра (Algebra). Один из ключевых аспектов для эффективной реализации высокопроизводительных задач. Библиотека EML включает базовые пакеты подпрограмм линейной алгебры 1, 2 и 3 уровней (BLAS1/2/3); стандартный пакет линейной алгебры LAPACK; пакеты для работы с разреженными матрицами (SPBLAS1/2/3).
- Сигнал (Signal). Различные функции для обработки сигналов, включая быстрые преобразования Фурье и Хартли, вычисление характеристик сигнала, свертка, фильтрация и усиление, генерация сигналов/шумов и т.п.

С точки зрения высокопроизводительных вычислений, именно математические функции, реализованные в библиотеках BLAS, LAPACK и FFT, являются основополагающими блоками многих научных и инженерных расчётов. Матричные операции, в том числе с одинарной и половинной точностью являются ключевыми компонентами многих актуальных алгоритмов анализа данных, включая задачи реализации искусственного интеллекта на базе машинного (глубокого) обучения. Актуальное состояние реализации математических функций в библиотеке EML приведено в таблице 1.

Таблица 1. Таблица готовности математических функций библиотеки EML по состоянию на 1 апреля 2019 года

Раздел библиотеки	Всего функций	Реализовано	Оптимизировано под Elbrus
Core	5	5	5
Vector	433	433	433
Algebra	464	464	360
Signal	167	167	167

Таким образом, несмотря на то, что математические функции LAPACK и SPBLAS сейчас не полностью оптимизированы, необходимые математические функции реализованы в библиотеке EML, и большая часть эффективно оптимизирована под Эльбрус. Оптимизация параллельной версии библиотеки на базе OpenMP и расширение функциональности по запросам пользователей находится в ближайших планах МЦСТ.

Стоит отметить, что библиотечные функции EML имеют собственные программные интерфейсы, которые довольно близки к Intel MKL и IPP. Также в состав EML включены переходники для полного повторения стандартных интерфейсов BLAS/LAPACK/FFTW.

3.2. Мультимедийные функции

Современные вычислительные задачи часто связаны также с обработкой мультимедийных данных (аудио- и видеосигналов), в связи с чем наличие эффективной реализации мультимедийных функций также обладает высоким приоритетом при оценке готовности программной экосистемы для НРС. В состав математической и мультимедийной библиотеки EML входят следующие оптимизированные мультимедийные функции.

- Изображение (Image). Актуальная версия библиотеки EML содержит оптимизированные функции для работы с изображениями, такие как создание/удаление, заполнение и копирование изображений, арифметические/логические операции с изображениями, преобразования, статистика, авто и кросс корреляция, фильтрация. Всего реализовано 113 функций.
- Видео (Video). Оптимизированные функции работы с видео, входящие в состав EML позволяют эффективно проводить дискретное косинусное преобразование, де/квантизацию, цветовые преобразования, полпиксельную интерполяцию и усреднение, изменять разрешение изображения и т.п. Всего реализовано 119 функций.

Также на Эльбрус портирована библиотека компьютерного зрения OpenCV [4].

Интересным примером использования мультимедийных функций библиотеки EML может являться недавно опубликованная работа по распознаванию двумерных предметов искусства с помощью быстрой реализации расстояния Хэмминга на платформе Эльбрус [5]. Для эффективной работы с памятью используются специальные методы префетчинга. Также на Эльбрус успешно реализована и работает система распознавания паспортов [6].

4. Опыт переноса на архитектуру Эльбрус некоторых пакетов для HPC

Долгое время суперкомпьютеры преимущественно использовались для решения научных и инженерных задач вычислительного моделирования. К настоящему времени для архитектур с традиционным набором команд (CISC) существует ряд прикладных программных пакетов для моделирования различных физических и химических процессов. Исследователями активно используются специализированные программные пакеты для моделирования химии и физики твёрдых сред, газов и жидкостей, атомных и молекулярных взаимодействий, биомолекулярного моделирования и анализа, астрофизической магнито-гидродинамики, систем электронов и ядер и т.п.

Наличие готовых пакетов, обеспечивающих хорошую производительность вычислений, позволяет исследователям из соответствующих научных областей проводить расчёты, требующих больших вычислительных затрат. Перенос основных программных пакетов для проведения высокопроизводительных расчётов позволит исследователям эффективно использовать суперкомпьютеры на базе архитектуры e2k, не вдаваясь в детали архитектурно-зависимой оптимизации.

4.1. Методы переноса пакетов на архитектуру Эльбрус

В настоящее время для переноса готовых пакетов прикладного ПО на архитектуру Эльбрус существует три подхода.

1. *Использование режима бинарной совместимости.* Для архитектуры Эльбрус доступны средства поддержки эффективной двоичной совместимости с широко распространенной архитектурной платформой Intel IA-32 (x86), x86-64 на базе динамической бинарной компиляции. При бинарной компиляции с двоичных кодов Intel x86 также выполняется автоматическая оптимизация кодов для архитектуры Эльбрус при использовании статической и динамической оптимизирующей компиляции. Реализованная для серии микропроцессоров «Эльбрус» система бинарной компиляции [7] имеет три уровня оптимизации – шаблонный транслятор (второй уровень), легкий оптимизатор (третий уровень), полноценный оптимизатор (четвертый уровень). Для обеспечения качественной и непрерывной работы эмулируемого кода применен ряд программно-аппаратных решений:

- оптимизация третьего и четвертого уровней производится на отдельном ядре микропроцессора;
- в случае одноядерного процессора оптимизация производится дискретным квантованным разделением времени с исполнением кода;
- используется аппаратная трансляция адресов доступа к памяти эмулируемой x86-машины;
- используется аппаратный буфер трансляции адресов исполняемого кода для эффективной поддержки операции передачи управления в эмулируемом коде;
- для ускорения сохранения x86-контекста заведены специализированные регистры;
- поддерживается многопоточная работа кодов x86.

Многие прикладные программы часто доступны только в кодах x86, что делает систему динамической бинарной трансляции крайне востребованной на практике. Тем не менее, данный режим может быть полезен только в начальной фазе переноса программ с Intel на Эльбрус, но на последующих этапах более оптимальной является перекомпиляция оптимизирующим компилятором.

2. Сборка из исходных кодов при помощи оптимизирующего компилятора. При компиляции программ для архитектуры Эльбрус, относящейся к классу VLIW, производительность получающегося оптимизированного кода определяется работой компилятора в большой степени, по сравнению с другими вычислительными архитектурами, и при этом важнейшими проблемами, которые должны решаться компилятором, являются:

- обнаружение параллелизма элементарных инструкций кода (операций);
- планирование операций в широкие команды, которые запускаются на исполнение одновременно в одном такте микропроцессора
- использование различных механизмов аппаратной поддержки производительности.

Также следует отметить, что в настоящее время доступна система кросс-сборки программных пакетов для дистрибутива ОС «Эльбрус» [8], которая позволяет в автоматическом режиме разрешать большинство проблем этапов сборки и конфигурирования. Основной проблемой этого этапа являются архитектурно-зависимые участки кода. В частности, в компилируемом коде могут использоваться ассемблерные вставки с реализацией той или иной функциональности. В большинстве случаев, тем не менее, существует так называемая generic-реализация, основанная на использовании стандартных системных библиотек, которую можно использовать для первичной сборки и доработать при необходимости получения более высокой производительности. При сборке можно использовать специальные ключи для повышения производительности. (см. пример ключей конфигурирования cmake при сборке OpenCV на рис. 1)

```
-O4 -ffast -ffast-math -faligned -D_LITTLE_ENDIAN -fPIC
-fno-ipo-region -fno-vir
```

Рис. 1. CMAKE_C_FLAGS для сборки OpenCV при помощи системы конфигурирования cmake

3. Наконец, можно провести ряд оптимизаций исходных кодов для повышения производительности кода при выполнении на архитектуре Эльбрус. Общепринятые методы ускорения вычислений эффективно работают и для архитектуры Эльбрус, поэтому они могут быть эффективно применены и для данной архитектуры. Таким образом, при оптимизации существующей библиотеки/программного пакета, необходимо:

- Провести профилирование, на основе результатов профилирования определить наиболее затратные по времени фрагменты кода
- Для вычислительных операций, требующих высокой производительности, использовать функции из библиотеки EML.
- При необходимости провести дополнительные оптимизации для оставшихся наиболее затратных по времени фрагментов кода
 - Изменение организации локальных данных, чтобы избежать непрямого адресации в «горячих» циклах.
 - Перемещение инвариантных выделений/освобождений памяти за пределы циклов.
 - Использование линеаризованных многомерных массивов.
 - Замена многократных вложенных вызовов функций на векторизированные версии.
 - Модификация подпрограмм, содержащих не прямые обращения в память на запись; замещаются на вложенные циклы с непрямыми обращениями на чтение.
 - Использование ключевого слова restrict.
 - Компиляция проекта с ключом -fwhole (аналог опции -fno для других компиляторов), что позволяет использовать inline для вызовов функций между модулями.

Таким образом, стандартный перенос любых пакетов на языках высокого уровня не вызывает никаких трудностей. Однако, задача оптимизации иногда требует значительных временных ресурсов и соответствующей квалификации.

4.2. Статус переноса пакетов на архитектуру Эльбрус

Ниже представлен список пакетов с открытым программным кодом, активно используемых для высокопроизводительных вычислений в различных отраслях наук. Перенесены и оптимизированы приложения газодинамики: NOISEtte [9, 10] и Tapir [11]. В процессе переноса:

- Биоинформатика: BLAST
- Молекулярная динамика: GROMACS, LAMPPS, NAMD
- Вычислительная химия: GAMESS
- Вычислительная гидродинамика: OpenFOAM
- Физика высоких энергий: GEANT4, MILC
- Погода и климат: NEMO, WRF

Упомянутые пакеты частично используются в бенчмарках производительности SPECсru2006, SPECсru2017. В частности, в данных бенчмарках есть задачи из пакетов GROMACS, NAMD, GAMESS, MILC, WRF, которые успешно работают на архитектуре Эльбрус и участвуют в оценке производительности, как в однопоточном, так и в многопоточном режимах исполнения, а также при оценке пропускной способности системы (rate). В связи с этим есть полная уверенность в том, что перенос вышеуказанных пакетов на платформу Эльбрус не должен быть очень трудоемким, а после небольшой настройки пакетов под особенности архитектуры они должны работать с высокой производительностью.

5. ПО для работы с данными

Неотъемлемой частью высокопроизводительных вычислений является работа с большими объемами структурированных и неструктурированных данных. При работе с данными в НРС требуется параллельный доступ, высокая пропускная способность, надёжность, высокая доступность и масштабируемость. Эти требования актуальны не только для задач обработки больших данных, искусственного интеллекта и машинного обучения, но и для классических высокопроизводительных задач математического моделирования.

В классификации ниже перечислено доступное ПО для работы с данными на архитектуре Эльбрус.

- Базы данных.
 - PostgreSQL [12] — свободная объектно-реляционная система управления базами данных (СУБД), базируется на языке SQL. Обладает высокой масштабируемостью. Периодически используется в научных высокопроизводительных приложениях, в связи с чем на некоторых вычислительных кластерах существуют выделенные узлы с установленной СУБД PostgreSQL, например в ЦОД NERSC [13]. Статус: Перенесена и оптимизирована под архитектуру Эльбрус. Также перенесены и оптимизированы СУБД MySQL [14] и SQLite [15].
- Параллельные системы хранения данных.
 - Lustre [16] — распределённая параллельная файловая система, одна из наиболее популярных файловых систем для высокопроизводительных вычислительных систем. Обеспечивает высокую пропускную способность и масштабируемость. Статус: Перенесена и оптимизирована под архитектуру Эльбрус.
 - CEPH [17]. Система CEPH позволяет строить объектную систему хранения данных распределённого типа и одновременно обеспечивать как файловый, так и блочный интерфейсы доступа. Доступность и надёжность системы обеспечивается встроенными механизмами репликации. Статус: Перенесена и оптимизирована под архитектуру Эльбрус.
- Фреймворки для технологии больших данных
 - Apache Hadoop [18] — это один из наиболее популярных фреймворков для работы с большими данными. Apache Hadoop включает свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих с большими данными с использованием модели MapReduce. Статус: Перенесен и оптимизирована под архитектуру Эльбрус.

Также существуют программно-аппаратные решения для работы с данными — серверы Эльбрус-4.4 БД [19] и Эльбрус-4.4 ЦЭФ [20], аналогичные серверы разрабатываются на процессорах Эльбрус-8С.

6. Заключение

Если сравнивать программную экосистему, разработанную для архитектуры Эльбрус, например с экосистемой, разработанной для архитектуры ARM [21], мы можем отметить что экосистема для архитектуры Эльбрус уже сейчас находится в достаточно зрелом состоянии, чтобы использоваться для решения высокопроизводительных задач научного и инженерного моделирования и обработки больших данных.

- Базисом программной экосистемы является наличие зрелого и эволюционирующего компилятора, а также оптимизированных научных библиотек.

- Для архитектуры Эльбрус разработаны компиляторы для современных версий основных распространенных языков (C, C++, Fortran, Java, JS, C#). Качество компиляторов (статических и бинарного) по надежности позволяет успешно компилировать с оптимизацией и исполнять сотни миллионов строк программных комплексов в самых различных прикладных сферах.
 - Математическая и мультимедийная библиотека EML содержит все необходимые функции, методы и примитивы для эффективного решения высокопроизводительных научных и инженерных задач.
 - В дистрибутиве ОС «Эльбрус» присутствуют необходимые системы конфигурирования (CMake, Autotools). Активно развивается система кросс-сборки программных пакетов для дистрибутива ОС «Эльбрус».
 - Ведутся активные работы по портированию на архитектуру Эльбрус программного обеспечения для работы с данными. Разработаны оптимизированные версии популярной СУБД PostgreSQL, распределённых параллельных файловых систем Lustre и CEPH.
- Необходимо расширение пользовательской экосистемы. По некоторым направлениям работы уже ведутся.
 - Наличие открытых репозиторий. Уже сейчас ОС «Эльбрус» для стандартных процессоров архитектуры x86 находится в свободном доступе для скачивания [22]. Появление репозиторий с открытыми исходными кодами как для самой ОС «Эльбрус», так и для доступных программных пакетов может обеспечить увеличение доступных программных библиотек под ОС «Эльбрус».
 - Создание учебных материалов. Примером может являться учебник [23], выпущенный в 2013-м году и описывающей результаты многолетней работы специалистов компании АО «МЦСТ».
 - Использование различных инструментов обучения, включая составление и проведение лекционных курсов, создание учебных центров и лабораторий, создание образовательных порталов, проведение олимпиад, конкурсов и хакатонов.
 - Большая интенсивность на профессиональных конференциях, включая проведение мастер-классов и курсов повышения квалификации.
 - Активность на интернет-площадках — поддержка в актуальном состоянии интернет портала АО «МЦСТ» [24], создание и модерирование пользовательских форумов, описание эффективных сценариев использования на популярных ресурсах (например, habr.com - в настоящее время 27 публикаций с тэгом «Эльбрус», stackoverflow.com).

Литература

1. Ким А. К., Перекатов В.И., Фельдман В.М. Центры обработки данных на базе серверов «Эльбрус» // Вопросы радиоэлектроники. 2017. No 3. С. 6–12. УДК 004.382.7 (075)
2. Ким А.К., Перекатов В.И., Фельдман В.М. На пути к российской экзасистеме: планы разработчиков аппаратно-программной платформы «Эльбрус» по созданию суперкомпьютера эксафлопсной производительности // Вопросы радиоэлектроники. 2018. No 2. С. 6–13. УДК 004.382.7 (75)
3. Ишин П.А., Логинов В.Е., Васильев П.П. Ускорение вычислений с использованием высокопроизводительных математических и мультимедийных библиотек для

- архитектуры Эльбрус // Вестник воздушно-космической обороны, М.: Научно-производственное объединение «Алмаз» им. акад. А.А. Расплетина, 2015, № 4 (8). 64–68.
4. Culjak I., Abram D., Pribanic T., Dzapo H., Cifrek M. A brief introduction to OpenCV // 2012 Proceedings of the 35th International Convention MIPRO, Opatija, 2012, P. 1725–1730.
 5. Limonova E., Skoryukina N., Neiman-zade M. Fast Hamming distance computation for 2D art recognition on VLIW-architecture in case of Elbrus platform // SPIE 11041, Eleventh International Conference on Machine Vision (ICMV 2018), 110411N (15 March 2019); doi: 10.1117/12.2523101; DOI: 10.1117/12.2523101
 6. МЦСТ. Новости. Технологии распознавания Smart Engines поддерживают архитектуру Эльбрус. URL: <http://www.mcst.ru/tekhnologii-raspoznaniya-smart-engines-podderzhivayut-arkhitekturu-elbrus> (дата обращения: 10.04.2019).
 7. Ким А.К., Бычков И.Н., Волконский В.Ю., Воробушков В.В., Груздов Ф.А., Михайлов М.С., Нейман-заде М.И., Парахин Ю.Н., Семенихин С.В., Слесарев М.В., Фельдман В.М. Российские технологии «Эльбрус» для персональных компьютеров, серверов и суперкомпьютеров. // Современные информационные технологии и ИТ-образование. Сборник докладов 9-й международной научно-практической конференции, Москва, 14-16 ноября 2014. С.39-49.
 8. Шалаев М.А. Сборка компонентов программного обеспечения вычислительных комплексов семейства «Эльбрус» // Вопросы радиоэлектроники. 2017. № 3. С. 39–43. УДК 004.4'2
 9. Абалакин И.В., Бахвалов П.А., Горобец А.В., Дубень А.П., Козубская Т.К., Параллельный программный комплекс NOISETTE для крупномасштабных расчетов задач аэродинамики и аэроакустики. // Вычислительные методы и программирование. Т.13. 2012. С. 110–125.
 10. Горобец А.В., Нейман-заде М.И., Окунев С. К., Калякин А. А., Суков С. А.: Производительность процессора Эльбрус-8С в суперкомпьютерных приложениях вычислительной газовой динамики // Препринты ИПМ им. М. В. Келдыша, 2018, 152, 20 с.
 11. Gorobets A., Soukov S., Bogdanov P. Multilevel parallelization for simulating turbulent flows on most kinds of hybrid supercomputers // Computers and Fluids, 2018, In Press. <https://doi.org/10.1016/j.compfluid.2018.03.011>
 12. PostgreSQL. URL: <https://www.postgresql.org/> (дата обращения: 11.04.2019).
 13. NERSC Data, Analytics & Services: PostgreSQL. URL: <https://www.nersc.gov/users/data-analytics/data-management/databases/postgresql/> (дата обращения: 11.04.2019).
 14. MySQL. URL: <https://www.mysql.com/> (дата обращения: 11.04.2019).
 15. SQLite. URL: <https://www.sqlite.org/index.html> (дата обращения: 11.04.2019).
 16. The Lustre file system. URL: <http://lustre.org/> (дата обращения: 11.04.2019).
 17. Ceph Home. URL: <https://ceph.com/> (дата обращения: 11.04.2019).

18. Apache Hadoop. URL: <https://hadoop.apache.org/> (дата обращения: 11.04.2019).
19. Сервер «Эльбрус-4.4 БД». URL: <http://www.ineum.ru/elbrus-4.4-bd> (дата обращения: 11.04.2019).
20. Сервер «Эльбрус-4.4 Цэф». URL: <http://www.ineum.ru/elbrus-4.4-ceph> (дата обращения: 11.04.2019).
21. Banchelli Gracia, F.F. [et al.]. Is Arm software ecosystem ready for HPC? // SC17: International Conference for High Performance Computing, Networking, Storage and Analysis. 2017.
22. Программное обеспечение «Эльбрус».
DOI: <http://mcst.ru/programmnoe-obespechenie-elbrus>
23. Ким А.К., Перекаатов В.И., Ермаков С.Г. Микропроцессоры и вычислительные комплексы семейства «Эльбрус». — СПб.: Питер, 2013. — 272 с.: ил. ISBN 978-5-459-01697-0
24. МЦСТ Эльбрус. Российские микропроцессоры и вычислительные комплексы.
URL: <http://www.mcst.ru/> (дата обращения: 10.04.2019).