

Раздел 1. Принципы построения и архитектура суперкомпьютеров

*С.М. Абрамов, С.А. Дбар, А.В. Климов, Ю.А. Климов,
А.О. Лацис, А.А. Московский, А.Ю. Орлов, А.Б. Шворин*

ВОЗМОЖНОСТИ СУПЕРКОМПЬЮТЕРОВ «СКИФ» РЯДА 4 ПО АППАРАТНОЙ ПОДДЕРЖКЕ В ПЛИС РАЗЛИЧНЫХ МОДЕЛЕЙ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ*

*Институт программных систем им. А.К. Айламазяна РАН,
г. Переславль-Залесский, Россия,
abram@botik.ru, moskov@lcc.chem.msu.ru, orlov@mccme.ru,
shvorin@gmail.com*

*Институт прикладной математики им. М.В. Келдыша РАН,
г. Москва, Россия,
ssa@kiam.ru, klimov@keldysh.ru, yuklimov@keldysh.ru, lacis@kiam.ru*

Введение. Преодоление петафлопсного рубежа производительности суперкомпьютеров, разработка решений транспетафлопсного уровня требуют использования десятков и сотен тысяч процессорных ядер, работающих в едином комплексе. Переход на новый уровень производительности и масштабируемости требует развития новых моделей параллельных вычислений, способов организации параллельных программ и новых архитектур.

В настоящее время наблюдается переход на гибридные, гетерогенные архитектуры суперкомпьютеров, узлы которых состоят из стандартных фон-неймановских процессоров и программируемых логических интегральных схем (ПЛИС). Среди них особенно перспективными являются архитектуры, в которых ПЛИС тесно связана с интерконнектом и используется и как маршрутизатор интерконнекта, и как средство поддержки базовых операций, и как ускоритель вычислений. Такие гибридные суперкомпьютеры семейства «СКИФ» ряда 4 разрабатываются по Программе Союзного государства СКИФ-ГРИД [1,4,11,12].

На базе этих суперкомпьютеров авторами ведется разработка методов поддержки различных моделей параллельных вычислений, методов и систем гибридного программирования на гетерогенных суперкомпьютерах, содержащих стандартные процессоры и ПЛИС. Гибридное программирование подразумевает, что приложение подразделяется на

* Работы выполняются при поддержке научно-технической программы Союзного государства «СКИФ-ГРИД», а также РФФИ по проекту № 09-07-13598-офи-ц

прикладную программу и библиотеки, исполняемые в стандартных процессорах, и на схемы, загружаемые в ПЛИС, выполняющие критические по времени исполнения части программы и операции нижнего уровня, в том числе связанные с обменом данными.

В целом, подходы к повышению производительности с помощью ПЛИС можно подразделить на два класса:

- 1) реализация в ПЛИС общесистемных операций нижнего уровня для конкретной модели вычислений; при этом одна загрузка в ПЛИС обслуживает все прикладные программы, выполняемые в данной системе программирования, а прикладные программы выполняются на стандартном процессоре;
- 2) реализация в ПЛИС критичных по эффективности фрагментов прикладных программ или их целиком; в этом случае для каждой прикладной программы загружается новая схема в ПЛИС.

В данном проекте ведутся работы по обоим направлениям с целью, в конечном счете, разработать для них общую базу.

Основные виды моделей параллельного программирования и наборы базовых операций, для которых проводится исследование и разработка новых средств поддержки в ПЛИС, следующие (упорядочены от широко распространенных до еще обрабатываемых в настоящее время):

- 1) массовые операции из MPI (редукции, барьер, бродкастинг и мультикастинг);
- 2) двусторонняя и односторонняя передача сообщений в стиле MPI2;
- 3) модель вычислений с разделенным глобальным адресным пространством (partitioned global address space, PGAS);
- 4) мелкотредовые и многопоточные модели вычислений, активные сообщения (active messages);
- 5) модели вычислений по готовности данных, с частично определенными данными, в стиле data-flow, и их дальнейшее развитие.

Перспективной целью является отработка универсальной базовой библиотеки нижнего уровня, которая может использоваться для реализации многих моделей вычислений, а также методов создания специализированных подсистем для конкретных моделей, систем и языков программирования.

Для отработки и апробации решений используются: открытая реализация MPI — MPICH2; интерфейсы и библиотеки нижнего уровня, используемые для PGAS-языков — GASNet и другие, а также разработанная в ИПС РАН T-система [2,3]. Применительно к MPICH2 и GASNet будут отработаны минимальные средства поддержки, а полный набор средств будет отработан на методах реализации T-системы и языков программирования высокого уровня на ее основе (таких как

параллельный Рефал Плюс [10]). Реализация Т-системы использует все основные современные механизмы параллельного программирования: и передачу сообщений, и глобальное адресное пространство (ссылок на Т-объекты), и средства быстрого запуска и выполнения мелких тредов, и вычисления по готовности данных, в том числе частично определенных.

В качестве средства реализации в ПЛИС критичных по эффективности фрагментов прикладных программ в ИПМ РАН разрабатывается специализированный язык Автокод HDL [5], предназначенный для программирования в конвейерном стиле, отображаемом в схемную реализацию в ПЛИС, ядер численных методов, имеющих небольшой объем кода (сотни строк).

Сопоставление с другими работами. За последние десятилетия в мире проводилось много работ по различным моделям вычислений и их аппаратной поддержке. Особенно большой опыт накоплен по поддержке базовых операций MPI в сетевых картах (Cray SeaStar, Quadrics, Mynet). Из нетривиальных моделей вычислений можно отметить модель вычислений data flow, которая за 30 лет прошла весь цикл жизни от больших ожиданий до разочарования. Однако подобные работы имеют смысл применительно к текущему уровню состояния аппаратных средств. При изменении технических характеристик параметры эффективности могут радикально меняться. В частности, мы полагаем, что на современном оборудовании с использованием ПЛИС может получить новое дыхание модель data flow (особенно в отечественном варианте — модель В.С. Бурцева [6] и ее развитие [9]) и другие модели, использующие ее элементы (Т-система [2,3], монотонные объекты [8] и другие). В данном проекте используется опыт как своих, так и сторонних разработок, но все решения привязываются к оборудованию суперкомпьютеров семейства «СКИФ» ряда 4. Из мировых работ можно отметить следующие.

1. Гибридные суперкомпьютеры, содержащие в каждом узле ускоритель вычислений с нестандартной архитектурой. Лидеры: суперкомпьютер IBM Roadrunner с сопроцессорами Cell и суперкомпьютер Cray XT5h с ПЛИС в качестве сопроцессоров. В обоих типах суперкомпьютеров сопроцессоры не имеют непосредственного выхода на интерконнект, что принципиально ограничивает их применение для поддержки параллельных вычислений. В нашем проекте разрабатываются методы аппаратной поддержки в ПЛИС распределенных вычислений совместно с реализацией в ПЛИС маршрутизатора интерконнекта.

2. Имеется много работ по аппаратной и программной поддержке в сетевых картах операций MPI. Для этой цели современные сетевые кар-

ты имеют средства программирования (Cray SeaStar, Quadrics, Myrinet). В нашем проекте мы разрабатываем средства поддержки, занимающих аналогичное положение в системах программирования, но для более перспективных моделей вычислений и на базе ПЛИС.

3. Модель вычислений PGAS — разделенное глобальное адресное пространство. Для PGAS уже хорошо отработаны интерфейсы нижнего уровня — GASNet и ARMCI, транслируемые в них языки программирования UPC (Unified Parallel C), CAF (Co-Array Fortran) и другие. В ведущих суперкомпьютерных фирмах IBM, Cray, Sun разработка суперкомпьютеров ведется с учетом поддержки PGAS. Самая интересная из архитектур — Cray XMT. Мы используем опыт этих разработок, поскольку элементы PGAS так или иначе входят и в другие модели вычислений.

4. Мелкотредовые и многопоточные модели вычислений развиваются с конца 1980-х – начала 1990-х годов. Имеется множество программных реализаций, дать обзор которых в короткой статье невозможно. Только в последнее время стали появляться компьютеры, специально поддерживающие такие модели, например, Cray XMT. Однако они еще не стали массовыми и поэтому дороги.

5. Программные реализации мелкотредовых и многопоточных моделей вычислений не показали себя практичными за редким исключением, к которым принадлежит система Charm++ Иллинойского университета и T-система ИПС РАН. Сильной стороной системы Charm++ является простота и минимальность понятий. T-система гораздо более богата высокоуровневыми понятиями (T-функции как мелкие треды, данные с глобальной адресацией, неопределенные данные), на основе которых реализован язык T++, обеспечивающий плавный переход от последовательного программирования на C++ к параллельному программированию. Эффективность и масштабируемость таких систем можно резко поднять при аппаратной поддержке в ПЛИС.

6. Средства схемного программирования ускорителя вычислений на базе ПЛИС сопоставимы со следующими зарубежными работами. VHDL и Verilog — стандартные средства программирования ПЛИС; они слишком низкоуровневые и непривычные для прикладного программиста-математика, поэтому на практике ими не пользуются. Handel C — императивный операторный язык с дискретным временем и единственностью источника сигнала. Его недостаток: нет возможности записать конвейер. Mitrion C — функциональный язык без побочных эффектов. Недостаток: слишком «рафинированная» модель вычислений, непривычная для массового программиста. Impulse C — представляет модель вычислений транспьютеров. Catapult C — имеет на входе чистый,

не аннотированный C++; проект выглядит нереалистичным. Практика использования перечисленных специализированных языков более высокого уровня, чем VHDL и Verilog, показывает, что быстродействие по сравнению с «ручным» программированием на языках VHDL и Verilog падает не менее чем в 5 раз.

7. Среди отечественных работ по схемным вычислениям лидирует проект реконфигурируемых многопроцессорных вычислительных систем НИИ МВС ЮФУ [7]. В нем реализован так называемый структурно-процедурный способ организации мультиконвейерных вычислений, одна из важнейших особенностей которого заключается в том, что приложение в целом автоматически разбивается транслятором на компоненты, которые исполняются на специальной системе из ПЛИС и процессоров. Это потребовало разработки большого специализированного языка Colato как языка высокого уровня для более продуктивного программирования, чем VHDL и Verilog, и создания качественно новой методики программирования.

В свою очередь, разрабатываемый в ИПМ РАН язык программирования схем Автокод HDL [5] занимает промежуточное положение между указанными в пп. 6 и 7 языков высокого и низкого уровня. С одной стороны, он сохраняет основные понятия низкого уровня от VHDL и Verilog, предоставляя возможность реализовывать эффективные схемы; с другой стороны, избавляет от многих технических деталей, которые не требуются для программирования ускорителей вычислений. Система понятий Автокода HDL доступна прикладному математику. При этом Автокод HDL не предназначен для кодирования больших динамически реконфигурируемых схем, на которые нацелен язык Colato.

Используемое оборудование. Методы разработки аппаратных средств поддержки моделей вычислений и систем программирования в значительной степени зависят от текущего уровня развития технологий и используемого оборудования. В суперкомпьютерах «СКИФ» ряда 4 задача сравнительно дешевой разработки современных суперкомпьютеров высшей производительности с использованием передовых технологий, а также проблема сочетания новых и классических архитектур, решается следующим образом.

Вычислительные узлы. В вычислительных узлах использована передовая серийная элементная база, вышедшая на рынок в 2009 году, а именно процессоры Intel® Xeon® 5570 (Nehalem) и соответствующий им чипсет, а также ПЛИС Altera EP4SGX230 и другие микросхемы. Схема вычислительного узла приведена на рис. 1, технические характеристики — в таблице 1.

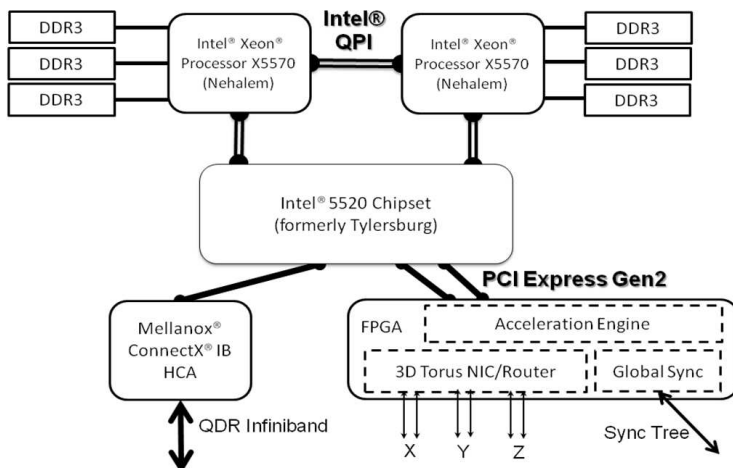


Рис. 1. Схема вычислительного узла суперкомпьютеров «СКИФ» ряда 4

Технические характеристики вычислительного узла и основных коммуникационных сетей

Таблица 1

Тип процессора	4-ядерный Intel® Xeon® CPU X5570 @ 2.93GHz
Количество процессоров	2
Пиковая производительность	93.76 Гфлоп/с (на 2 процессора узла)
Тип сопроцессора	ПЛИС Altera EP4SGX230
Топология системной сети	Трехмерный тор (3D-тор)
Максимальная скорость передачи данных системной сетью (3D-тор)	60 Гбит/с/узел (суммарная на 6 линках)
Задержка передачи сообщений на уровне прикладных программ (MPI)	1–2 мкс
Тип вспомогательной сети	Mellanox ConnectX InfiniBand HCA QDR
Максимальная скорость передачи данных вспомогательной сетью InfiniBand	40 Гбит/с/узел
Дисковая память	Накопитель SSD 80 Гбайт
Оперативная память	12 Гбайт

Сеть. Основное «узкое место» современных суперкомпьютеров — взаимодействие вычислительных узлов между собой: пересылка сообщений, синхронизация, коллективные операции, доступ к памяти других узлов и т. п. Производительность коммуникационной сети (интерконнекта), определяемая пропускной способностью и задержкой, лимитирует производительность современных суперкомпьютеров. Ведущие зарубежные фирмы, такие как IBM и Cray, для сохранения своего лидерства на рынке суперкомпьютеров разрабатывают проприетарные сети, которые доступны только в поставке дорогостоящих суперкомпьютеров их собственной разработки. Более того, самые производительные сети часто находятся под эмбарго.

В суперкомпьютерах «СКИФ» ряда 4 принято следующее конкурентоспособное современное решение: суперкомпьютеры содержат две основные коммуникационные сети:

1. Вспомогательная сеть на базе серийного сетевого оборудования Mellanox ConnectX Infiniband HCA QDR. Эта сеть обеспечит стандартные режимы работы, доступ к параллельной файловой системе суперкомпьютера, функционирование средств управления суперкомпьютером. Максимальная пропускная способность этой сети — 40 Гбит/с/узел.

2. Системная сеть на базе программируемых логических интегральных схем (ПЛИС) с линками, объединенными в сеть с топологией 3D-тор. Эта сеть поддержит разработку новых эффективных методов маршрутизации, реализацию операций, связанных с обменом, новых моделей параллельных вычислений и т. п. Максимальная пропускная способность этой сети в полтора раза больше вспомогательной — 60 Гбит/с/узел.

Гибридная архитектура. Таким образом, в суперкомпьютерах «СКИФ» ряда 4 принята гибридная архитектура. Это означает, что:

1. Подмножество аппаратных средств составляет суперкомпьютер классической архитектуры: вычислительные узлы на базе процессоров Intel, совместимых с x86_64 архитектурой; интерконнект (вспомогательная сеть) на базе серийного сетевого оборудования QDR Infiniband и т. д.

2. Дополнительно вычислительные узлы включают в себя программируемые логические интегральные схемы (ПЛИС) самого высокого на настоящий момент уровня интеграции Altera Stratix IV GX 230, которые могут выступать в разных ролях:

- в качестве сопроцессора — ускорителя вычислений;
- в качестве маршрутизатора системной сети с топологией 3D-тор;
- в смешанных качествах.

Это дает уникальную (на настоящий момент) возможность поддержки в ПЛИС как новых моделей параллельных вычислений, так и старых систем программирования, особенно в части эффективной коммуникации и доступа к распределенной памяти.

Текущее состояние работ. Охарактеризуем ход работ и ближайшие планы по разработке средств аппаратной поддержки в ПЛИС моделей вычислений и систем программирования на суперкомпьютерах «СКИФ» ряда 4.

1. Экспериментальное исследование оборудования, подбор решений низкого уровня.

Проведено экспериментальное исследование методов взаимодействия процессоров и ПЛИС на стендовом оборудовании суперкомпьютеров СКИФ ряда 4, состоящем из одного узла, для выявления наиболее выгодных режимов работы при передаче активных сообщений, односторонних операций с удаленной памятью, запуска мелких тредов и других операций. Для этого были разработаны специальные тесты на языке VHDL, загружаемые в ПЛИС, и языке C, выполняемые в процессоре. Взаимодействие процессор–ПЛИС осуществляется через интерфейс PCI Express. Для передачи данных из процессора в ПЛИС используется программная запись по адресам, настроенных на ПЛИС через BAR (base address register). Для приема данных из ПЛИС используется либо прямая запись из ПЛИС в оперативную память, либо спинлок (spin lock) — ожидание поступления данных в цикле чтения. Над этими механизмами надстраиваются прием и передача сообщений через очереди — циклические буфера, ожидание активных сообщений и т. д. При этом маршрутизатор сети, реализованный в ПЛИС, оптимизируется под эффективную передачу (с низкой задержкой и высокой частотой сообщений) коротких сообщений, характерных для обрабатываемых в данном проекте моделей вычислений. Выявлены режимы работы, обеспечивающие передачу из процессора в ПЛИС и обратно около 20 млн 64-байтовых сообщений в секунду. Это хороший показатель для реализации различных моделей вычислений.

2. Разработка средств аппаратной поддержки MPI и PGAS.

На базе маршрутизатора сети с топологией трехмерного тора в ПЛИС [12] и реализации MPI на базе MPICH2, реализуемых в рамках проекта суперкомпьютеров СКИФ ряда 4, разработаны дополнительные механизмы поддержки односторонних операций доступа к памяти для эффективной реализации модели вычислений PGAS и библиотеки shmem. Исследуется вопрос поддержки языков типа PGAS — Unified Parallel C (UPC) и Co-Array Fortran (CAF). На UPC, CAF и на языках C и Фортран с использованием библиотеки shmem программируются и экспериментально исследуются тесты и модельные задачи с нерегулярным

доступом к памяти (задачи на неструктурных адаптивных сетках). В качестве первой модельной задачи реализован на односторонних записях сеточный аналог задачи Дирихле на неструктурированной сетке. Ведется работа над реализацией теста NAS UA (Unstructured Adaptive) на односторонних операциях MPI2 и shmem.

3. Разработка набора примитивов низкого уровня для реализации data-flow-моделей и T-системы.

Более сложные механизмы аппаратной поддержки отрабатываются на моделях вычислений, близких к data flow, и на разработанной в ИПС РАН T-системе [2,3]. Базовые понятия T-системы включают в себя средства запуска и останова мелких тредов по готовности/неготовности данных, глобальную адресацию распределенных объектов и другие специальные понятия. Исследуется вопрос, какие из этих средств оправдано поддерживать в ПЛИС и маршрутизаторе, а какие имеет смысл реализовывать, как и раньше, программно в стандартном процессоре. Для реализации первых версий аппаратной поддержки T-системы достаточно базовых механизмов, используемых и в других моделях: активные сообщения, запуск мелких тредов по готовности данных, односторонний доступ к удаленной памяти. В дальнейшем представляет большой интерес аппаратная поддержка специфических понятий T-системы, таких как глобальная адресация распределенных объектов, «T-суперструктура» и др.

4. Разработка методов схемных вычислений в ПЛИС и языка Автокод HDL.

Язык схемного программирования Автокод HDL [5] предназначен для реализации в ПЛИС критичных по эффективности фрагментов прикладных программ. Язык строится как двухуровневый. На сегодня устоялся, реализован и документирован базовый уровень, который, собственно, и формирует модель программирования. Перечислим его основные понятия. Объекты, которыми манипулирует схема — скалярные и векторные регистры, а также векторные блоки двухпортовой синхронной памяти. Схема включает в себя один процесс, выполняющийся в дискретном времени, и комбинационные соединения, включая вставку компонентов. Отдельно задаются действия на каждом такте и действия по начальному сбросу. Предусмотрены векторные присваивания. Логика трехзначная ('0', '1', 'Z'). В комбинационной части можно записывать мультиплексоры, логику "when". Действует правило единственности источника значения. Есть циклы времени компиляции. В исполняемой части предусмотрены операторы отложенного перехода. Блоки памяти доступны через стандартную четверку регистров: адрес, данные на запись, данные чтения, разрешение записи. Предусмотрен стандартный интерфейс связи с внешним миром: два 32-разрядных регистра и диапа-

зон адресов синхронной памяти. Разработаны библиотечные компоненты работы с плавающей точкой. Интенсивно развивается второй уровень языка, призванный многократно снизить трудоемкость и объем кода при реализации приложений вычислительного характера. Разработано средство автоматического построения схем для вычисления сложных арифметических выражений, включая условную логику, в конвейерном режиме. Выполнен перенос Автокода HDL на платформу Скиф-Аврора.

Проектирование второго уровня языка велось на примерах простых модельных задач (численное решение уравнения теплопроводности). В качестве первой нетривиальной модельной задачи на Автокоде HDL реализована в ПЛИС задача Римана о распаде разрыва в линейном приближении. При расчете с двойной точностью в одном узле (на 1 ПЛИС Altera EP4SGX230) эта задача показывает реальную производительность около 20 Гфлоп/с с двойной точностью. Это приложение обгоняет два процессора узла более чем в 4 раза и при этом имеет заметный резерв как по частоте, так и по используемой площади кристалла. (Трудозатраты — около 2 человеко-недель. Объем исходных формул математической постановки задачи — 2 листа А4. Объем исходного текста на Автокод HDL — 6КБ, объем сгенерированного кода на VHDL — 150КБ.)

1. Абрамов С.М. Суперкомпьютерные технологии России: объективные потребности и реальные возможности // Международная научная конференция «Параллельные вычислительные технологии» (ПаВТ 2010), Уфа, 29 марта – 2 апреля 2010 г.
2. Абрамов С.М., Адамович А.И., Коваленко М.Р. Т-система — среда программирования с поддержкой автоматического динамического распараллеливания программ. Пример реализации алгоритма построения изображений методом трассировки лучей // Программирование, 1999. Т. 25 (2). С. 100–107.
3. Abramov S.M., Adamovich A.I., Inyukhin A.V., Moskovsky A.A., Roganov V.A., Shevchuk E.V., Shevchuk Yu.V. and Vodomerov A. OpenTS: An Outline of Dynamic Parallelization Approach // V. Malyshkin (ed.), Parallel Computing Technologies, 8th International Conference, PaCT 2005, Krasnoyarsk, Russia, September 5–9, 2005. Proceedings. LNCS 3606, 2005. P. 303–312.
4. Абрамов С.М., Анищенко В.В., Криштофик А.М., Парамонов Н.Н. Первые технические решения суперкомпьютерного направления «СКИФ» Ряда 4 // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской на-

- учной конференции (21–26 сентября 2009 г., г. Новороссийск). М.: Изд-во МГУ, 2009. С.161–165.
5. Андреев С.С., Дбар С.А., Лацис А.О., Плоткина Е.А. Система программирования Автокод HDL и опыт ее применения для схемной реализации численных методов в FPGA // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской научной конференции (21–26 сентября 2009 г., г. Новороссийск). М.: Изд-во МГУ, 2009. С. 237.
 6. Бурцев С.В. Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ: Сборник ст. М: Изд-во «ТОРУС ПРЕСС», 2006. 414 с.
 7. Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. Реконфигурируемые мультиконвейерные вычислительные структуры / Под общ. ред. И.А. Каляева. Ростов-на-Дону: Издательство ЮНЦ РАН, 2008. 320 с.
 8. Климов Анд. В. Детерминированные параллельные вычисления с монотонными объектами // Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ: Труды Всероссийской научной конференции (24–29 сентября 2007 г., г. Новороссийск). М.: Изд-во МГУ, 2007. С. 212–217.
 9. Климов Арк. В., Окунев А.С., Степанов А.М. Проблемы развития модели вычислений с управлением потоком данных и особенности ее архитектурной реализации // Материалы международной конференции «Информационные технологии в науке, образовании, телекоммуникации и бизнесе» (IT+S&E'07), Ялта–Гурзуф, май 2007 г.
 10. Климов Ю.А., Орлов А.Ю. Параллельное программирование на языке Рефал // Научный сервис в сети Интернет: решение больших задач: Труды Всероссийской научной конференции (22-27 сентября 2008 г., г. Новороссийск). М.: Изд-во МГУ, 2008. С. 241-243.
 11. Московский А.А., Перминов М.П., Соколинский Л.Б., Черепеников В.В., Шамакина А.В. Опыт использования суперкомпьютера «СКИФ-Аврора» для решения научно-технических задач // Международная научная конференция «Параллельные вычислительные технологии» (ПаВТ 2010), Уфа, 29 марта – 2 апреля 2010 г.
 12. Орлов А.Ю., Шворин А.Б. О реализации в ПЛИС маршрутизатора высокопроизводительной сети // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской научной конференции (21–26 сентября 2009 г., г. Новороссийск). М.: Изд-во МГУ, 2009. С. 208–210.