

# Архитектура и программное обеспечение пространственно-распределённых вычислительных систем<sup>\*)</sup>

В. Г. Хорошевский, М. Г. Курносов, С. Н. Мамойленко, А. Ю. Поляков

Рассматриваются архитектурные принципы и программное обеспечение пространственно-распределённых вычислительных систем. Описывается конфигурация пространственно-распределённой мультикластерной вычислительной системы, созданная и развиваемая Центром параллельных вычислительных технологий ГОУ ВПО “СибГУТИ” совместно с Институтом физики полупроводников им. А.В. Ржанова СО РАН. Приводится описание программного обеспечения системы, включающее средства, разработанные коллективом ведущей научной школы НШ-5176.2010.9.

*Ключевые слова:* пространственно-распределённые вычислительные системы, программное обеспечение.

## 1. Введение

Современный этап развития вычислительной индустрии и телекоммуникационных технологий [1] характеризуется построением пространственно-распределённых вычислительных систем (ВС) как макроколлективов рассредоточенных вычислительных устройств, взаимодействующих между собой через локальные и/или глобальные сети (включая сеть Internet). Первая пространственно-распределённая вычислительная система АСТРА была разработана Институтом математики СО АН СССР и Новосибирским электротехническим институтом МВ и ССО РСФСР в 1970 – 72 гг.; среди конфигураций ВС были районные, городские и междугородные.

В середине 1980-х годов идея объединения пространственно-рассредоточенных вычислительных средств получила воплощение в технологии метакомпьютинга (Metacomputing), предложенной исследователями из Национального центра суперкомпьютерных приложений США [2]. С появлением и бурным развитием всемирной сети Internet всё большую популярность получает технология использования компьютеров пользователей, позволяющая формировать среды для распределённых вычислений (Distributed or Peer-To-Peer Computing). В начале 1990-х годов [3] для описания пространственно-распределённых ВС стал применяться термин GRID (Global Resource Information Distribution) как метафора о такой же лёгкости доступа к вычислительным ресурсам, как и к электрической сети (Power Grid).

В данной работе описываются архитектурные принципы пространственно-распределённых ВС, их программное обеспечение и конфигурация системы, созданной Центром параллельных вычислительных технологий (ЦПВТ) Государственного образовательного учреждения высшего профессионального образования “Сибирский государственный университет телекоммуникаций и информатики” (ГОУ ВПО “СибГУТИ”) совместно с Лабора-

<sup>\*)</sup> Работа выполнена в рамках междисциплинарного проекта № 113 СО РАН и при поддержке РФФИ (гранты № 08-07-00018, 09-07-00095, 08-07-00022) и Совета по грантам Президента РФ (ведущая научная школа НШ-5176.2010.9).

торией вычислительных систем учреждения Российской академии наук Института физики полупроводников им. А.В. Ржанова Сибирского отделения РАН (ИФП СО РАН) [4]. Архитектура данной ВС основывается на результатах ведущей научной школы РФ по распределённым вычислительным системам и параллельному мультипрограммированию (НШ-5176.2010.9).

## 2. Архитектура пространственно-распределённых вычислительных систем

Функциональная структура пространственно-распределённой ВС представляется как композиция множества сосредоточенных вычислительных средств и сети связи [1]. Предельным по простоте вариантом сосредоточенного вычислительного средства является элементарная машина – композиция вычислительного модуля и системного устройства. Конфигурация вычислительного модуля допускает варьирование в широких пределах: от процессорного ядра до композиции, включающей универсальные и специализированные процессоры (например, GPGPU). Следовательно, простейшим вариантом пространственно-распределённой ВС является совокупность однотипных ЭМ, использующая “длинные” (программно-коммутируемые) каналы межмашинных связей.

К развитым вариантам сосредоточенных вычислительных средств относятся массово-параллельные ВС. Широкое применение в качестве базовых средств для формирования пространственно-распределённых ВС получили появившиеся в конце XX века вычислительные кластеры (кластерные ВС). В последнем случае пространственно-распределённые ВС принято называть мультикластерными.

Любое функциональное взаимодействие между сосредоточенными вычислительными средствами, составляющими пространственно-распределённую ВС, осуществляется через сеть связи. В сеть связи включаются каналы и системные устройства, распределённые по сосредоточенным вычислительным средствам. В пространственно-распределённых ВС нет принципиальных ограничений на физическую природу каналов; в них могут использоваться глобальные сети и сеть Internet. Именно поэтому такие системы и называют пространственно-распределёнными, т.е. объединяющими географически удалённые друг от друга сосредоточенные вычислительные средства.

Основу архитектуры пространственно-распределённых ВС составляют следующие принципы:

- *параллельность функционирования сосредоточенных вычислительных средств* (т.е. способность нескольких или всех сосредоточенных вычислительных средств совместно и одновременно решать одну сложную задачу, представленную параллельной программой);
- *независимость функционирования сосредоточенных вычислительных средств* (каждое вычислительное средство способно функционировать как автономно, так и в составе пространственно-распределённой ВС);
- *программируемость структуры* (т.е. возможность программировать сеть связи между сосредоточенными вычислительными средствами);
- *гомогенность состава* (т.е. программная совместимость различных сосредоточенных вычислительных средств);
- *открытость взаимодействий* (т.е. реализация строго определённого интерфейса взаимодействий между сосредоточенными вычислительными средствами).

К архитектурным свойствам пространственно-распределённых вычислительных систем можно отнести:

- *универсальность* (пространственно-распределённые ВС являются алгоритмически универсальными; в них могут быть реализованы любые алгоритмы, доступные сосредоточенным вычислительным средствам);

- *масштабируемость и производительность* (ВС не имеют принципиальных ограничений в наращивании производительности; увеличение суммарной производительности достигается за счёт включения в систему новых сосредоточенных вычислительных средств);
- *реконфигурируемость* (сеть связей между сосредоточенными вычислительными средствами допускает автоматическую настройку);
- *большемасштабность* (количество сосредоточенных вычислительных средств в пространственно-распределённых ВС теоретически не ограничено);
- *живучесть* (в ВС все исправные сосредоточенные вычислительные средства доступны для совместного использования при реализации параллельных программ);
- *надёжность* (пространственно-распределённая ВС способна продолжать функционировать в случае отказов отдельных сосредоточенных вычислительных средств и обеспечивать заданный уровень производительности);
- *самоконтроль и самодиагностика* (сама организация живучего функционирования ВС связана с контролем правильности работы системы и с локализацией неисправностей в ней; контрольно-диагностические функции выполняются совместно всеми сосредоточенными вычислительными средствами);
- *техничко-экономическая эффективность* (реализация архитектурных принципов позволяет резко сократить сроки конфигурирования ВС, облегчает создание программного обеспечения систем и упрощает их техническую эксплуатацию).

Сосредоточенные вычислительные средства пространственно-распределённой ВС называют узлами.

### **3. Программное обеспечение пространственно-распределённых вычислительных систем**

Эффективность функционирования пространственно-распределённых ВС во многом определяется возможностями программного обеспечения. Можно выделить следующие типы программных компонентов, необходимых для организации эффективного функционирования пространственно-распределённых систем.

*Сетевая операционная система ВС.* Это базовый компонент, осуществляющий управление процессами вычислительных узлов ВС и предоставляющий программный интерфейс для доступа к аппаратуре. В настоящее время доминирующее положение занимает операционная система (ОС) GNU/Linux. В коммерческих системах с массовым параллелизмом применяются и узкоспециализированные операционные системы, ориентированные на организацию параллельных вычислений и, как правило, основанные на ядре Linux. Примерами могут служить ОС [5 – 8]: Cray Catamount, IBM Blue Gene Compute Node Kernel (CNK), ZeptoOS, Kitten.

*Средства разработки параллельных программ.* Параллельные программы для распределённых ВС преимущественно разрабатываются в модели передачи сообщений (Message passing), в рамках которой параллельные ветви программы взаимодействуют между собой путём обменов информационными сообщениями по каналам межмашинных связей. Для создания таких параллельных программ применяются коммуникационные библиотеки, например, библиотеки стандарта MPI (MPICH2 [9], OpenMPI [10]), PVM. Использование таких библиотек подразумевает явное обращение в программе к функциям передачи сообщений от одного процесса другим. Наряду с коммуникационными библиотеками, относящимся к низкоуровневым средствам создания параллельных программ, существуют высокоуровневые языки параллельного программирования. Эти языки не требуют от разработчика явного обращения к функциям межмашинных обменов и позволяют создавать программы в абстрактной модели ВС, например, в модели разделённой глобально адресуемой общей памяти (PGAS – Partitioned Global Address Space). Представителями языков программирования, реализующих эту модель, являются [11 – 16] IBM X10, Cray Chapel, Berkeley Titanium, Berkeley Unified Parallel C, Coarray Fortran, Sun Fortress.

Достаточно часто вычислительные узлы распределённых ВС являются многопроцессорными системами с общей памятью. По этой причине получил распространение подход, основанный на создании гибридных параллельных MPI-OpenMP программ – взаимодействие между ветвями программы, расположенными на разных вычислительных узлах, осуществляется средствами коммуникационной MPI-библиотеки, а в рамках одного узла создаются параллельные потоки, взаимодействующие через его общую память.

*Средства отладки и анализа эффективности параллельных программ.* Для отладки параллельных MPI- и OpenMP-программ применяются отладчики [17 – 22] TotalView, Allinea DDT, Valgrind, Intel Parallel Debugger, GNU gdb. Анализ параллельных MPI-программ подразумевают сбор и обработку информации о взаимодействии ветвей параллельной программы по каналам межмашинных связей ВС и формирование статистики вызовов коммуникационных функций. Для OpenMP-программ осуществляется оценка накладных расходов, связанных с синхронизацией и дисбалансом времени выполнения параллельных потоков. Пакеты анализа эффективности параллельных программ позволяют оценить коэффициент накладных расходов параллельной программы и выявить наиболее вычислительно-сложные участки программы. Наибольшее распространение получили следующие пакеты анализа эффективности параллельных программ [23 – 28]: Pallas Vampir, VampirTrace, Intel Trace Analyzer and Collector, TAU, KOJAK, mpiP.

*Средства системы пакетной обработки заданий.* Выполнение заданий – параллельных программ пользователей на распределённых ВС осуществляется в режиме пакетной обработки. В таком режиме пользователь формирует паспорт задачи, в котором указываются требуемые для выполнения её параллельной программы: количество процессорных ядер, объем оперативной памяти в узлах и др. После чего задача ставится в очередь. Порядок обслуживания задач в очереди и их запуск на решение определяются алгоритмами планирования (поддерживаемыми системой пакетной обработки заданий). На интерфейс для средств пакетной обработки заданий существует отраслевой стандарт POSIX 1003.2d. Наибольшее распространение получили системы [29 – 32] TORQUE, Altair PBS Pro, Oracle Grid Engine, IBM LoadLeveler. Перечисленные системы допускают подключение к ним внешних планировщиков, таких как Maui.

*Средства контроля и диагностики вычислительных ресурсов.* Использование колоссальных потенциальных возможностей современных распределённых ВС безусловно определяется их надёжностью и способами организации функционирования. Несмотря на высокую надёжность микроэлектроники, вероятность возникновения отказов в распределённых ВС повышается с ростом количества вычислительных узлов. Следовательно, особую актуальность приобретает организация отказоустойчивого функционирования распределённых ВС. Последнее требует присутствие программных компонентов, обеспечивающих контроль и диагностику функционирования системы. Наибольшее распространение получили свободно распространяемые пакеты мониторинга состояния вычислительных ресурсов [33 – 36]: Ganglia, Nagios, MonALISA, CeMon.

*Системы организации отказоустойчивого выполнения параллельных программ.* Одним из основных средств к организации отказоустойчивого выполнения параллельных программ на ВС являются контрольные точки восстановления (КТВ). Во время выполнения параллельной программы периодически осуществляется сохранение состояния ветвей на носителе информации. В случае отказа одной из ветвей вычислительный процесс восстанавливается из последней КТВ. Как правило, средства создания КТВ интегрируются с системами параллельного программирования и коммуникационными библиотеками. Можно выделить следующие проекты создания КТВ [37, 38]: BLCR – Berkeley Lab Checkpoint/Restart и DMTCP – Distributed MultiThreaded CheckPointing.

*Программное обеспечение промежуточного уровня GRID-систем.* Для организации функционирования пространственно-распределённых ВС, объединяющих географически рассредоточенные ВС, служит программное обеспечение промежуточного уровня GRID-систем (GRID middleware). Рассредоточенные ВС, входящие в состав пространственно-

распределённой ВС, могут функционировать под управлением различного программного обеспечения (разные операционные системы, системы пакетной обработки заданий и пр.). Для организации функционирования рассредоточенных ВС как единого комплекса необходимо решить следующие задачи:

- 1) обеспечить глобальную аутентификацию (Single sign-on) пользователей на всех узлах (сосредоточенных вычислительных средствах);
- 2) реализовать возможность безопасной передачи информации с единой точкой аутентификации между узлами;
- 3) представить возможность запуска задач пользователей на любом узле;
- 4) обеспечить диспетчеризацию задач пользователей с учётом загрузки узлов и каналов связи между ними.

Первые три задачи успешно решаются существующими пакетами (GRID Middleware) Globus Toolkit, gLite, UNICORE [39 – 41]. Для обеспечения диспетчеризации задач пользователей с учётом загрузки подсистем и каналов связи между ними служат GRID-диспетчеры (GRID Metascheduler, GRID Broker): GridWay, Common Scheduler Framework, Grid Service Broker, Moab Grid Scheduler, EGEE Workload Manager Service, Nimrod/G, MP Synergy, Condor-G [42].

## **4. Пространственно-распределённая мультикластерная вычислительная система ЦПВТ СибГУТИ**

### **4.1. Архитектура мультикластерной ВС**

Центром параллельных вычислительных технологий ГОУ ВПО “СибГУТИ” совместно с Лабораторией вычислительных систем ИФП СО РАН создана и развивается пространственно-распределённая мультикластерная вычислительная система (GRID-модель) [43]. Действующая конфигурация ВС (рис. 1, май 2010 г.), включает более 250 процессорных ядер и имеет пиковую производительность несколько TeraFLOPS. Система объединяет 9 пространственно-рассредоточенных вычислительных кластеров, причём кластеры А-Г расположены в серверных помещениях ЦПВТ ГОУ ВПО “СибГУТИ” (центр г. Новосибирска), а кластеры Н, I – в Лаборатории вычислительных систем ИФП СО РАН (Академгородок, СО РАН).

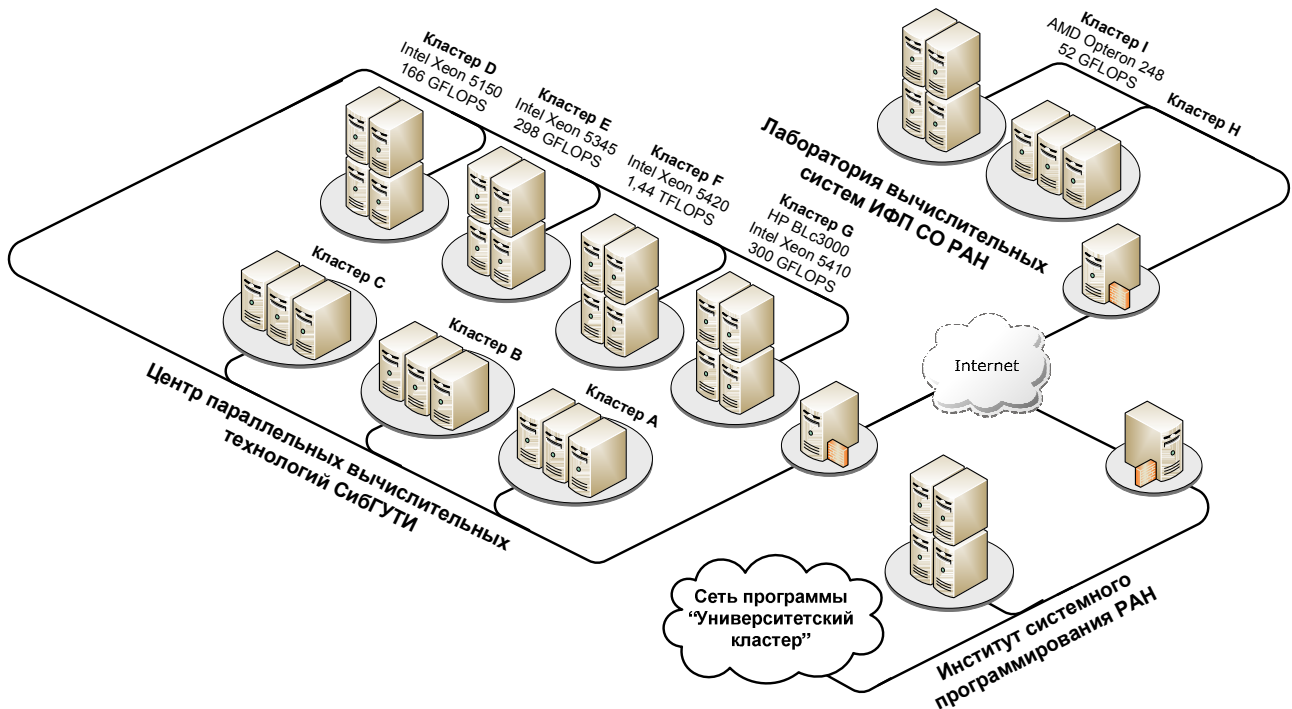
Системы А, В, С и Н являются кластерами рабочих станций и персональных компьютеров, которые располагаются в учебных лабораториях Кафедры вычислительных систем ГОУ ВПО “СибГУТИ” и в Лаборатории вычислительных систем ИФП СО РАН. Коммуникационные сети кластеров А, В, С и Н построены на базе технологии Gigabit Ethernet.

Кластеры D, E, F, G и I построены на базе двухпроцессорных узлов и функционируют в круглосуточном режиме. Каждый из этих кластеров укомплектован: вычислительными узлами и управляющим узлом, вычислительной и сервисной сетями связи, а также системами бесперебойного электропитания. Кластер D (Xeon16) объединяет 4 вычислительных узла, на каждом из которых размещено по два двухъядерных процессора Intel Xeon 5150 (Woodcrest) с тактовой частотой 2.66 GHz. Пиковая производительность кластера D - 166 GFLOPS. Кластер E (Xeon32) включает 4 вычислительных узла с двумя процессорами Intel Xeon E5345 (Quad-core) с тактовой частотой 2.33 GHz. Производительность кластера E – 298 GFLOPS. Кластер F (Xeon80) укомплектован 18 вычислительными узлами с двумя процессорами Intel Xeon E5420 (Quad-core) с тактовой частотой 2.5 GHz. Производительность кластера E – 1.44 TFLOPS. Кластер I оснащён 5 вычислительными узлами с двумя процессорами AMD Opteron 248 – 252 (Sledgehammer) с тактовой частотой 2.2 – 2.6 GHz. Пиковая производительность кластера – 52 GFLOPS. Актуальная информация о конфигурации системы представлена на сайте ЦПВТ ГОУ ВПО “СибГУТИ” [43].

В 2009 году ЦПВТ ГОУ ВПО “СибГУТИ” стал участником программы “Университетский кластер” [44], в рамках которой компания Hewlett-Packard передала Центру вычисли-

тельный кластер на базе блейд-серверов (кластер G), укомплектованный 4 HP ProLiant BL220c, на которых размещено по два четырёхядерных процессора Intel Xeon E5410 с тактовой частотой 2.33 GHz. Производительность кластера G в целом – 300 GFLOPS.

Любой из кластеров способен функционировать как автономно, так и в составе распределённой ВС. Коммуникационные среды вычислительных кластеров построены на базе технологий Gigabit и Fast Ethernet. Для объединения кластеров используется сеть Internet (технология VPN). Мультикластерная ВС допускает масштабирование путём организации взаимодействия с множеством других систем.



**Рис. 1.** Конфигурация пространственно-распределённой мультикластерной вычислительной системы (май 2010 года)

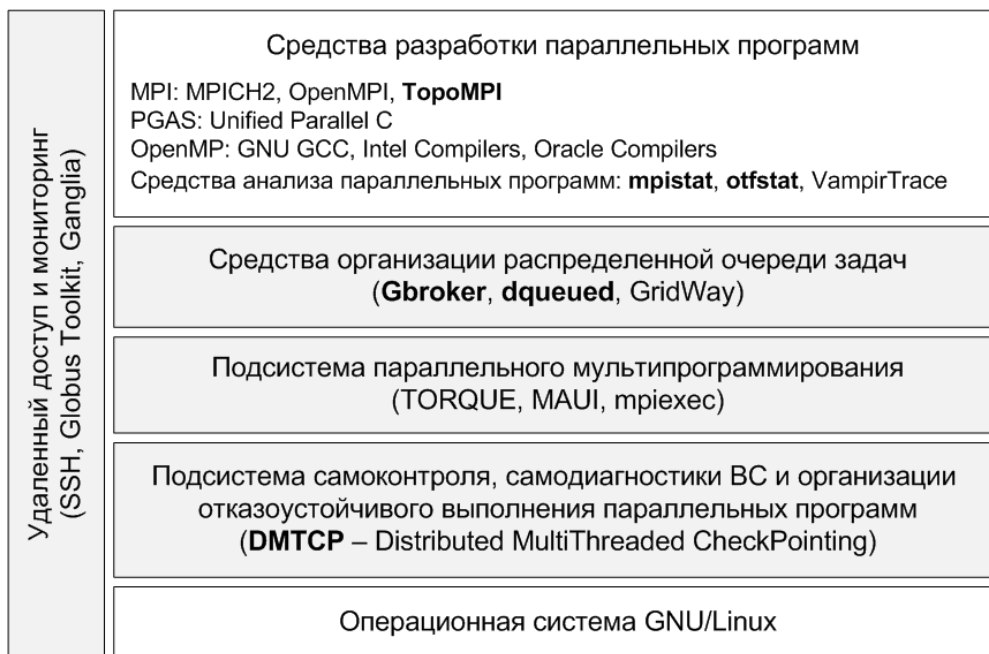
## 4.2. Программное обеспечение мультикластерной ВС

Пространственно-распределённая мультикластерная вычислительная система функционирует под управлением свободно распространяемого и созданного коллективом научной школы члена-корреспондента РАН В.Г. Хорошевского (НШ-5176.2010.9) программного обеспечения (рис. 2). Среди свободно распространяемых компонентов программного обеспечения имеются:

- операционная система GNU/Linux (дистрибутивы CentOS и Fedora);
- компиляторы с языков программирования C/C++, Fortran проекта GNU, а также соответствующие компиляторы компаний Intel и Oracle;
- библиотеки стандарта MPI: MPICH2, OpenMPI;
- средства отладки и анализа параллельных программ: VampirTrace, ompP;
- компоненты управления очередями задач: TORQUE, MAUI;
- система мониторинга Ganglia; пакет создания контрольных точек восстановления параллельных программ DMTCP;
- программное обеспечение Globus Toolkit промежуточного уровня GRID-систем; диспетчер задач GridWay.

Программные компоненты, созданные коллективом школы НШ-5176.2010.9:

- подсистема организации функционирования ВС в мультипрограммных режимах, включающая средства вложения параллельных программ и реализации эффективных коллективных обменов информацией между ветвями параллельных MPI-программ;
- средства анализа параллельных MPI-программ;
- пакеты организации функционирования распределённых очередей задач;
- средства самоконтроля, самодиагностики и создания контрольных точек восстановления параллельных программ.



**Рис. 2.** Программное обеспечение пространственно-распределённой мультикластерной ВС

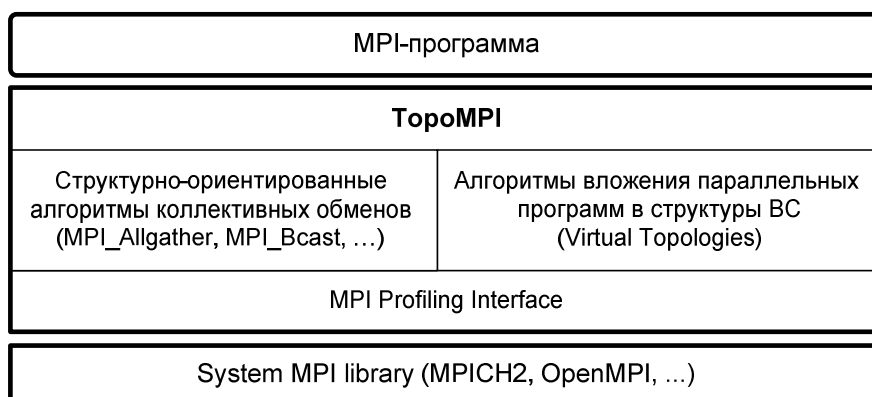
Рассмотрим некоторые из созданных программных компонентов организации функционирования пространственно-распределённых мультикластерных ВС.

*Библиотека ТороMPI коммуникационных функций.* Библиотека ТороMPI [45, 46] – это библиотека коммуникационных функций, ориентированная на эффективную реализацию параллельных программ на современных мультиархитектурных распределённых ВС с иерархической структурой. В состав библиотеки (рис. 3) входят структурно-ориентированные алгоритмы коллективных обменов информацией между ветвями параллельных программ и средства вложения параллельных программ в структуры распределённых ВС.

В библиотеке используется интерфейс профилирования стандарта MPI для перехвата обращений к функциям коллективных операций информационных обменов, остальные вызовы передаются системной библиотеке MPI (MPICH2, OpenMPI и др.). Описание иерархической организации распределённой ВС загружается библиотекой из конфигурационного файла при инициализации MPI-программы. На основе информации о распределении ветвей по ядрам и машинам системы определяется количество выделенных программе вычислительных узлов. При создании коммуниторов осуществляется оптимизация алгоритмов коллективных обменов и вложение графа программы в структуру распределённой ВС.

*Пакет DMTCP создания контрольных точек восстановления программ.* Для обеспечения отказоустойчивости параллельных программ используется пакет DMTCP (Distributed MultiThreaded Checkpointing), этот пакет реализован на уровне системных библиотек и является универсальным координирующим распределённым средством создания контрольных точек. Пакет DMTCP разработан и развивается Северо-западным университетом (Northeastern University) США. Коллектив ЦПВТ принимает участие в работах по его усовершенствованию. В частности, реализованы алгоритмы координированного восстановления набора

связанных процессов, восстановления взаимосвязей типа “родитель – потомок” и принадлежности к группам и сеансам для набора процессов в рамках одного узла вычислительной системы [47, 48]. Также ведутся работы по интеграции пакета DMTCP с системой для вычислений высокой пропускной способности Condor [49]. Предложен подход к реверсивной отладке на базе DMTCP [50].



**Рис. 3.** Функциональная структура библиотеки ToroMPI

## 5. Заключение

Пространственно-распределённая мультикластерная вычислительная система – инструментальное средство для отработки архитектурных решений и GRID-технологий, для отладки инструментальных средств параллельного мультипрограммирования и обеспечения живучести. Она активно используется в процессе подготовки специалистов в области параллельных вычислительных технологий.

Перспективы использования созданного инструментального средства в народном хозяйстве определяется растущими потребностями в применении распределённых гетерогенных вычислительных и GRID-систем (в областях, в которых требуются высоконадёжные (живучие) ВС, где решаются суперсложные задачи и моделируются современные технологические процессы и природные явления).

## 6. Литература

1. Хорошевский В.Г. Архитектура вычислительных систем. – М.: МГТУ им. Н.Э. Баумана, 2008. – 520 с.
2. Short History of Metacomputer: Past to Present, сайт NCSA, <http://archive.ncsa.illinois.edu/Cyberia/MetaComp/MetaHistory.html> (дата обращения 12.04.2010).
3. Computational Grids., I. Foster, C. Kesselman. Chapter 2 of "The Grid: Blueprint for a New Computing Infrastructure", Morgan-Kaufman, 1999. URL: <http://www.globus.org/alliance/publications/papers.php> (дата обращения: 12.04.2010)
4. Хорошевский В.Г., Мамойленко С.Н., Курносков М.Г. Архитектурные концепции, анализ и организация функционирования вычислительных систем с программируемой структурой// Труды Международной научно-технической конференции “Информационные технологии и математическое моделирование систем”. – М.: “Радиотехника”, 2009, с.74-79. ISBN: 978-5-88070-230-5
5. Cray XT Programming Environment User's Guide. URL: <http://docs.cray.com/books/S-2396-21/S-2396-21.pdf> (дата обращения 12.04.2010).



6. IBM System Blue Gene Solution: Blue Gene/P Application Development.  
URL: <http://www.redbooks.ibm.com/redbooks/pdfs/sg247287.pdf> (дата обращения 12.04.2010).
7. ZeptoOS: The small Linux for big computers. URL:  
<http://www.mcs.anl.gov/research/projects/zeptoos/> (дата обращения 12.04.2010)
8. Kitten Lightweight Kernel. URL: <https://software.sandia.gov/trac/kitten> (дата обращения 12.04.2010)
9. MPICH2: High-performance and Widely Portable MPI.  
URL: <http://www.mcs.anl.gov/research/projects/mpich2> (дата обращения 12.04.2010)
10. Open MPI: Open Source High Performance Computing. URL: <http://www.open-mpi.org> (дата обращения 12.04.2010)
11. X10 Programming Language. URL: <http://x10-lang.org> (дата обращения 12.04.2010)
12. Chapel Programming Language Homepage: Overview. URL: <http://chapel.cray.com> (дата обращения 12.04.2010)
13. Titanium Project Home Page. URL: <http://titanium.cs.berkeley.edu> (дата обращения 12.04.2010)
14. Berkeley Unified Parallel C (UPC) Project. URL: <http://upc.lbl.gov> (дата обращения 12.04.2010)
15. Co-Array Fortran. URL: <http://www.co-array.org> (дата обращения 12.04.2010)
16. The Fortress Language Specification. URL: <http://research.sun.com/projects/plrg/fortress.pdf> (дата обращения 12.04.2010)
17. TotalView Technologies – Linux Debugger and Memory Leak Detection Tool for Multi-Threaded C/C++ and Fortran. URL: <http://www.totalviewtech.com> (дата обращения 12.04.2010)
18. Allinea DDT. URL <http://www.allinea.com> (дата обращения 12.04.2010)
19. Valgrind Home. URL: <http://valgrind.org> (дата обращения 12.04.2010)
20. Intel® Debugger 9.1. URL:  
[http://www.intel.com/cd/00/00/21/92/219299\\_intel\\_debugger.pdf](http://www.intel.com/cd/00/00/21/92/219299_intel_debugger.pdf) (дата обращения 12.04.2010)
21. Intel® Parallel Composer – расширение для отладки параллельных программ. URL:  
<http://software.intel.com/ru-ru/articles/intel-parallel-composer> (дата обращения 12.04.2010)
22. GDB: The GNU Project Debugger. URL: <http://www.gnu.org/software/gdb> (дата обращения 12.04.2010)
23. VAMPIR. URL: <http://www.vampir.eu> (дата обращения 12.04.2010)
24. VampirTrace. URL: [http://www.vampir.eu/flyer/Flyer\\_VampirTrace\\_SC09.pdf](http://www.vampir.eu/flyer/Flyer_VampirTrace_SC09.pdf) (дата обращения 12.04.2010)
25. Intel® Trace Analyzer и Intel® Trace Collector версии 7.2 для Linux или Windows  
URL: <http://www.intel.com/cd/software/products/emea/rus/379896.htm> (дата обращения 12.04.2010)
26. TAU – Tuning and Analysis Utilities. URL:  
<http://www.cs.uoregon.edu/research/tau/home.php> (дата обращения 12.04.2010)
27. KOJAK. URL: [www.fz-juelich.de/zam/kojak](http://www.fz-juelich.de/zam/kojak) (дата обращения 12.04.2010)
28. mpiP: Lightweight, Scalable MPI Profiling. URL: <http://mpip.sourceforge.net> (дата обращения 12.04.2010)
29. Cluster resources :: Products – TORQUE Resource Manager.  
URL: <http://www.clusterresources.com/> (дата обращения 12.04.2010)
30. PBS Works – Enabling On-Demand Computing. URL: <http://www.openpbs.org> (дата обращения 12.04.2010)

31. Grid Computing | Oracle Grid Engine | Software | Sun Microsystems. URL: <http://www.sun.com/software/sge> (дата обращения 12.04.2010)
32. IBM Redbooks | Workload Management with LoadLeveler. URL: <http://www.redbooks.ibm.com/abstracts/sg246038.html> (дата обращения 12.04.2010)
33. Ganglia Monitoring System. URL: <http://ganglia.sourceforge.net> (дата обращения 12.04.2010)
34. Nagios – The Industry Standard in IT Infrastructure Monitoring. URL: <http://www.nagios.org> (дата обращения 12.04.2010)
35. MonALISA. URL: <http://monalisa.caltech.edu> (дата обращения 12.04.2010)
36. CEMon. URL: <http://vdt.cs.wisc.edu/components/cemon.html> (дата обращения 12.04.2010)
37. Paul H. Hargrove and Jason C. Duell Berkeley Lab Checkpoint/Restart (BLCR) for Linux Clusters In Proceedings of SciDAC 2006: June 2006.
38. J. Ansel, K. Arya, G. Cooperman, DMTCP: Transparent Checkpointing for Cluster Computations and the Desktop // Proc. of IEEE International Parallel and Distributed Processing Symposium (IPDPS'09) . IEEE Press, 2009.
39. The Globus Alliance. URL: <http://www.globus.org> (дата обращения 12.04.2010)
40. EGEE Technical: Middleware. URL: <http://technical.eu-egee.org/index.php?id=149> (дата обращения 12.04.2010)
41. UNICORE – Distributed computing and data resources. URL: <http://www.unicore.eu> (дата обращения 12.04.2010)
42. Grid Brokers and Metaschedulers Market Overview. URL: <http://www.gridwisetech.com/metaschedulers> (дата обращения 12.04.2010)
43. Вычислительные ресурсы Центра параллельных вычислительных технологий ГОУ ВПО “СибГУТИ” // Сайт ЦПВТ ГОУ ВПО “СибГУТИ”, URL: <http://cpct.sibsutis.ru/wiki/index.php/Main/Resources> (дата обращения: 12.04.2010).
44. Институт системного программирования РАН – программа “Университетский кластер”, URL: <http://www.ispras.ru/ru/unicluster/> (дата обращения: 12.04.2010).
45. Курносов М.Г. Алгоритмы вложения параллельных программ в иерархические распределённые вычислительные системы // Вестник СибГУТИ. – 2009. – № 2 (6). – С. 20-45.
46. Khoroshevsky V., Kurnosov M. Mapping Parallel Programs into Hierarchical Distributed Computer Systems // Proceedings of 4th International Conference “Software and Data Technologies (ICSOF 2009)”. – Sofia: INSTICC, 2009. – Vol. 2. – P. 123-128.
47. Поляков А.Ю. О восстановлении программ из контрольной точки // Параллельные вычислительные технологии (ПаВТ’2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.) [Электронный ресурс] – Челябинск: Издательский центр ЮУрГУ, 2010. – С. 299-310.– ISBN 978-5-696-03987-9. URL: <http://omega.sp.susu.ac.ru/books/conference/PaVT2010/full/155.pdf>
48. Поляков А.Ю. Об алгоритме восстановления структуры процессов из контрольной точки // Материалы Международной научно-технической конференции “Многопроцессорные вычислительные и управляющие системы (МВУС-2009)”.– Таганрог: ТТИ ЮФУ.– 2009.– Т. 2.– С. 71–73.– ISBN 978-5-8327-0341-1.
49. G. Cooperman, K. Arya, P. Keller, A.Y. Polyakov DMTCP and Condor: a New Checkpointing Mechanism // Condor Week 2010. – Режим доступа: <http://www.cs.wisc.edu/condor/CondorWeek2010/condor-presentations/cooperman-dmtcp.pdf>
50. Ana Maria Visan, Artem Polyakov, Praveen S. Solanki, Kapil Arya, Tyler Denniston, Gene Cooperman Temporal Debugging using URDB .– Режим доступа: <http://arxiv.org/abs/0910.5046v1>

*Статья поступила в редакцию 04.05.2010*

**Хорошевский Виктор Гаврилович**

Член-корреспондент Российской академии наук, профессор, доктор технических наук, заслуженный деятель науки Российской Федерации, заведующий Лабораторией вычислительных систем Института физики полупроводников им. А.В. Ржанова СО РАН, директор Центра параллельных вычислительных технологий, заведующий Кафедрой вычислительных систем Сибирского государственного университета телекоммуникаций и информатики. Область научных исследований – архитектура распределённых вычислительных систем (ВС), параллельное мультипрограммирование, методы организации функционирования ВС в моно- и мультипрограммных режимах, методы анализа надежности, живучести и технико-экономической эффективности функционирования большемасштабных ВС, осуществимость параллельного решения задач.

Тел.&факс: (383) 333-21-71, 269-82-75; e-mail: khor@isp.nsc.ru, khor@sibsutis.ru

**Курносков Михаил Георгиевич**

Кандидат технических наук, доцент Кафедры вычислительных систем Сибирского государственного университета телекоммуникаций и информатики, младший научный сотрудник лаборатории вычислительных систем Института физики полупроводников им. А.В. Ржанова СО РАН. Область научных исследований: модели и методы организации функционирования большемасштабных распределённых вычислительных систем, системное программное обеспечение.

Тел.&факс: (383) 333-21-71, 269-82-75; e-mail: mkurnosov@gmail.com.

**Мамойленко Сергей Николаевич**

Доцент, к.т.н., заместитель заведующего Кафедрой вычислительных систем Сибирского государственного университета телекоммуникаций и информатики. Область научных исследований – параллельное мультипрограммирование, методы организации функционирования вычислительных систем в моно- и мультипрограммных режимах.

Тел. (383) 269-82-75, e-mail: msn@sibsutis.ru, msn@isp.nsc.ru.

**Поляков Артём Юрьевич**

Младший научный сотрудник Лаборатории вычислительных систем Института физики полупроводников им. А.В. Ржанова СО РАН, старший преподаватель Кафедры вычислительных систем Сибирского государственного университета телекоммуникаций и информатики. Область научных исследований – отказоустойчивость выполнения параллельных программ, самоконтроль и самодиагностика ВС.

Тел.&факс (383) 333-21-71, 269-82-75, e-mail: artpol84@gmail.com.

**Architecture and software of geographically-distributed computer systems**

**V. G. Khoroshevsky, M. G. Kurnosov, S. N. Mamoienko, A.Yu. Polyakov**

Architecture principles and software of geographically-distributed computer systems are described. A real configuration of geographically-distributed multicluster computer system is presented. This system has been created and developed by the Computer centre for parallel technologies of Siberian State University for Telecommunication technologies and Information techniques in conjunction with A.V. Rzhzanov Institute of semiconductor physics of Siberian Branch of Russian Academy of Science. The description of the system software (including open source components and extensions developed by collective of the author's scientific school on distributed computer systems) is resulted.

*Keywords:* geographically-distributed computer systems, system software.