

УДК 004.75 + 62-5
ББК 32.973

**ГРАФОДИНАМИЧЕСКИЕ СИСТЕМЫ
С СЕТЕЦЕНТРИЧЕСКИМ УПРАВЛЕНИЕМ
В МАТЕМАТИЧЕСКИ ОДНОРОДНОМ ПОЛЕ
КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ**

Затуливетер Ю. С.¹, Фищенко Е. А.²
*(Учреждение Российской академии наук
Институт проблем управления РАН, Москва)*

В развитие идей графодинамики обсуждаются проблемы воплощения в глобальной компьютерной среде систем с сетецентрическим управлением высокой структурной сложности. На основе исчисления древовидных структур предлагается подход к формированию в ресурсах глобальных сетей математически однородного поля компьютерной информации, в котором открываются возможности представления и «бесшовного» программирования графодинамических систем с сетецентрическим управлением в едином формализме. Приводятся примеры систем с сетецентрическим управлением, которые можно отнести к классу графодинамических.

Ключевые слова: компьютерные сети, сетецентрическое управление, древовидные структуры, графодинамика, единое адресное пространство, интеграция.

¹ Юрий Семенович Затуливетер, ведущий научный сотрудник, кандидат технических наук, доцент (zvt@ipu.rssi.ru, Москва, ул. Профсоюзная, д. 65, тел. 8-(495) 334-92-09).

² Елена Алексеевна Фищенко, ведущий научный сотрудник, кандидат технических наук (fish@ipu.rssi.ru, Москва, ул. Профсоюзная, д. 65, тел. 8-(495) 334-92-09).

1. Введение

За прошедшие 20 лет большие системы пополнились принципиально новым классом объектов исследований. Компьютерные сети образовали качественно новую среду – глобальное информационное пространство, которое связывает более 1,5 млрд. персональных компьютеров. Их количество быстро растет. Сети мобильной связи объединяют более 4 млрд. абонентов.

Очевидно, «центр тяжести» компьютерной среды от изначально разрозненных локальных компьютерных/сетевых архитектур и приложений смещается к глобальным сетям и массовым приложениям, интегрированным в общемировое информационное пространство. Сверхбольшие системы этого класса быстро эволюционируют от слабосвязных к сильносвязным. Их особенность в том, что связность систем универсально программируемых вычислительных устройств воплощается универсальными компьютерными сетями, системообразующие свойства которых инвариантны к количеству и составу связываемых устройств. Архитектура больших сильносвязных систем этого класса становится сетевидной, поэтому процессы управления их функционированием и развитием, воплощаемые в них, могут рассматриваться как сетевидные.

Образцы сфер применимости таких систем. Все более актуальной для практики становится концепция «Интернет вещей» [11]. Распределенные системы на основе массовых сетей, в состав которых могут входить подсети интеллектуальных датчиков и всевозможных исполнительных устройств, способны обеспечивать управление разнообразными системами объектов со встроенными компьютерными устройствами.

Миллиарды встроенных компьютерных устройств обеспечивают управление сложными объектами и системами стационарных и подвижных объектов во все более разнообразных сферах массовых применений: в различных областях производства, в системах управления автомобилями, в авиации, в управлении транспортными и товарными потоками, в интеллектуализации бытовой техники и др.

Функционирование и развитие мировой социосистемы в условиях глобального информационного пространства во все более существенной мере становится зависимым от свойств компьютерной среды. В осязаемой перспективе глобальная компьютерная среда должна стать универсальным стратегическим ресурсом и одним из главных инструментов сетецентрического управления функционированием и устойчивым развитием мировой социосистемы в целом и ее частей [3, 4].

Совокупный функциональный и вычислительный потенциал глобальной сети с географически отдаленными компьютерами открывает практически неограниченные перспективы массового применения разнообразных систем с сетецентрическим управлением. Однако общедоступное применение распределенного компьютерно-сетевого интеллекта, отвечающее требованиям массового спроса, пока невозможно. Выделим две причины, относящиеся, на наш взгляд, к числу основных:

- отсутствие средств «бесшовного» программирования в ресурсах глобальных сетей распределенных структурно сложных вычислений;
- отсутствие универсальных математических методов постановки и решения задач управления системами распределенных объектов, для которых характерна высокая структурно-динамическая сложность, в условиях непрерывно меняющегося информационного контекста глобальных компьютерных сетей.

Для преодоления этих препятствий требуется решение проблем фундаментального характера, которые связаны как с разработкой теоретических и прикладных методов решения в условиях глобальной компьютерной среды задач сетецентрического управления высокой структурно-динамической сложности, так и с построением инструментальных средств их практического воплощения. Для этого необходима принципиальная коррекция классической компьютерной аксиоматики [3, 4].

2. Графодинамические системы

Высокая структурная сложность больших систем с управлением, погружаемых в сетевую среду, объясняется быстро нарастающим производством и потреблением все более разнообразных и сложных информационных структур, которые сопровождают функционирование и развитие социосистемы в целом и ее частей в глобальном информационном пространстве [4]. Для формализации таких систем требуются новые принципы и методы, позволяющие оперировать не только со скалярами или регулярными структурами типа вектор или матрица, но и со сложными данными, имеющими произвольно изменяющиеся структуры, которые, в общем случае, зависят от быстро меняющегося контекста глобального информационного пространства. Отсутствие универсального математического формализма и методов постановки и решения таких задач управления становится важнейшим сдерживающим фактором расширения сфер и масштабов их применения.

Известно, что до настоящего времени для математического и компьютерного решения задач используются существенно отличающиеся формализмы. Десятилетиями развитие компьютерных языков и формальных систем шло в направлении повышения уровней абстракции. Поначалу в рамках развития процедурных языков программирования доминировали попытки внешнего подражания математическим нотациям (Фортран, Алгол). На этом фоне также возникали непроцедурные компьютерные формализмы высоких уровней абстракции – Лисп, Пролог и др. В настоящее время атрибутом систем индустриального программирования стал объектно-ориентированный стиль.

Несмотря на обилие разнообразных компьютерных языков и систем программирования, понятийный барьер между ними и математическими формализмами/методами решения остается слишком высоким. По большому счету математические и компьютерные понятийные пространства постановки/решения задач по-прежнему представляют собой трудно совмещаемые культуры.

В условиях быстро прогрессирующей компьютерной глобализации информационного пространства новые реализации больших систем различного назначения обретают качества продуктов массового применения. Нетрудно видеть, что подавляющая доля таких систем, функционирующих в глобальном информационном пространстве, разрабатывается в основном коллективами разнопрофильных программистов. Такие продукты, как правило, строятся эвристическими методами, в отсутствие математических моделей с доказанными свойствами корректного (безопасного) функционирования.

Очевидно, чем больше масштабы применения таких решений, тем выше риски непредсказуемых последствий, связанных и с неизбежными для эвристических моделей ошибками в управлении ответственными системами и объектами, а также в нежелательных социальных последствиях массовых «сервисов», запускаемых без учета всей совокупности значимых факторов.

Проблемы изначальной семантической несовместимости математических и компьютерных формализмов обретают особую остроту применительно к большим системам с сетецентрическим управлением. Глобальные масштабы, реальное время, особая значимость ошибок и нестыковок, неизбежных для комплексных эвристических решений, заставляют искать пути кардинального снижения понятийных барьеров между математическими и компьютерными пространствами решений.

Путь к этому – формирование единого формализма математического и компьютерного решения задач сетецентрического управления в больших системах. В таком формализме открываются возможности построения математически корректных моделей и их непосредственного компьютерного воплощения в ресурсах глобальной компьютерной среды.

Полагаем, что отправной методологической идеей для общего подхода к математическому решению задач сетецентрического управления большими системами глобально распределенных объектов должны стать принципы графодинамики [1], обозначившие новое направление в расширении сфер применимости математического аппарата теории управления.

В работе [1] в сферы теоретического рассмотрения были впервые введены графодинамические системы. Это новый, и потому малоизученный класс систем с управлением. Отличительные особенности графодинамических систем состоят в том, что их текущие входы, выходы и внутренние состояния представляются структурами общего вида и могут принимать значения в произвольно задаваемых множествах графов, см. рис. 1. Развитие событий во времени [1] связано не с движениями по графам, а их изменением. Изменения могут быть либо автономными, либо вызваны внешними для системы воздействиями и задаются функциями преобразования графов. Преобразование графов может сопровождаться изменением состава вершин, связей между вершинами, а также значений «разметок», связываемых с вершинами и ребрами (дугами) графов.

Новые идеи математической формализации динамических систем, высказанные в основах графодинамики, дают начальный импульс для построения конструктивных теорий графодинамических систем с управлением. К сожалению, эта опередившая свое время работа не получила теоретического развития.

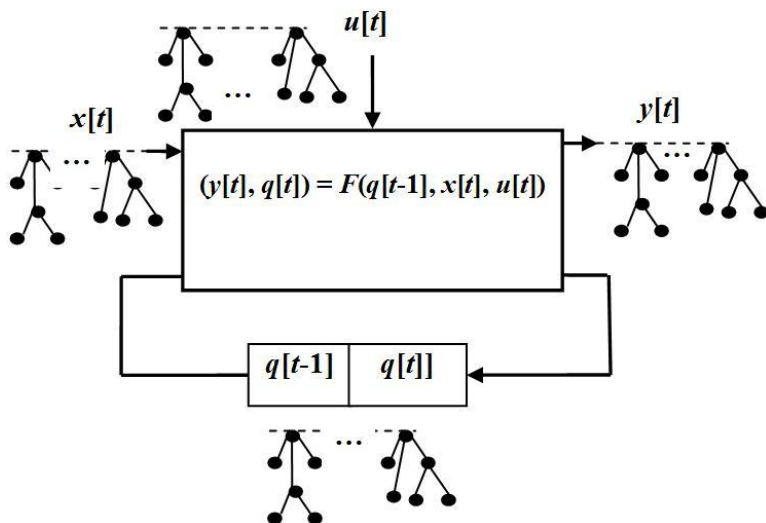


Рис. 1. Графодинамические системы

Причины, скорее всего, в прежнем отсутствии предметных областей со значимыми объектами приложения. Практическая потребность в математической теории графодинамических систем становится очевидной только сейчас, в связи с состоявшимся де-факто формированием глобального компьютерно-сетевого информационного пространства и массовым вовлечением в него социальной среды [4]. Ее функционирование и развитие в глобальном информационном пространстве сопровождается экспоненциально растущими потоками и объемами сложно структурированной информации, отсюда необходимость единого формализма решения и программирования задач управления в сверхбольших системах, в которых объектом математических преобразований становятся глобально распределенные динамически изменяемые структуры высокой сложности.

Масштабы и темпы роста спроса на системы распределенной обработки в сетевых ресурсах глобально распределенной информации в разнообразных сферах превышают возможности существующих технологий индустриального программирования. Наряду с развитием теории распределенных графодинамических систем совершенно необходимо построение общедоступных, математически и технологически «бесшовных» инструментов их программирования в ресурсах глобально-связной компьютерной среды.

Актуальность данной проблематики растет в темпах, соответствующих экспоненциальному росту потоков и объемов распределенной информации и глобальных масштабов применений компьютерных сетей.

К важным аспектам новизны данной работы можно отнести определение перспективных сфер применения графодинамических систем с сетецентрическим управлением, а также методов и средств «бесшовного» их программирования, основанных на исчислении древовидных структур. Это математически замкнутое компьютерное исчисление реализовано в системе программирования ПАРСЕК [5] и может рассматриваться как первый компьютерный базис для практически значимого воплощения формализма графодинамики.

3. Задачи распределенной обработки

Увеличение масштабов применения глобальных сетей сопровождается ростом разнообразия задач распределенной обработки данных. В таблице 1 приведены примеры таких систем, дающие представление о масштабах и сферах их применения.

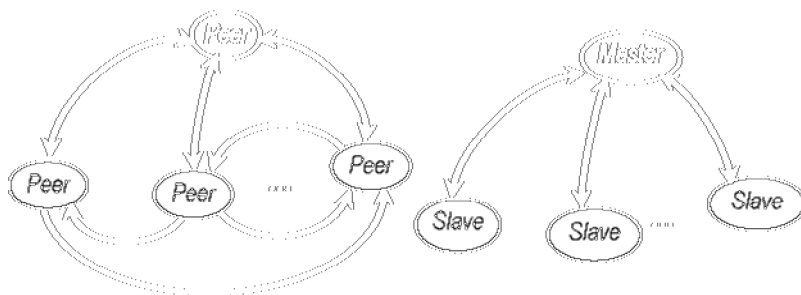
Таблица 1. Примеры распределенных систем обработки

Название	Цель	Средство	Цена
<i>Future Combat System (FCS), US Army</i> [12]	Соединить всех участников боевых действий и военную технику в единую командную сеть сетецентрического управления	-	\$177 млрд.
<i>LHC</i> (Большой Адронный Коллайдер), <i>CERN</i>	Обработка 1,5 Петабайт ежегодно на 10000 компьютерах	<i>GRID</i> , 100 организаций в 31 стране	~\$500 млн.
«Связьинвест» (Россия)	Обслуживание 40 млн. абонентов проводных сетей связи	Интегрированная биллинговая система	\$480 млн.

Она иллюстрирует противоречивые качества этих систем. Массовость вовлечения потребителей говорит о потенциально высоком спросе на системы глобально распределенной обработки, но высокая стоимость отражает чрезмерную трудоемкость их разработки и сопровождения.

В отсутствие «бесшовных» технологий универсального программирования глобально распределенных вычислений спрос на задачи распределенной обработки и управления с нарастающими темпами опережает предложение.

В настоящее время системы распределенной обработки строятся в сетевых архитектурах «Peer-to-Peer» (*P2P*), см. рис. 2, а, и «клиент-сервер», см. рис. 2, б.



а) Общий случай: все узлы равноправны, все узлы общаются друг с другом (равноправная структура)

б) Частный случай: обобщенный и централизованный лидер (клиент-серверная структура)

Рис. 2. Типы сетевых архитектур распределенной обработки

В первом случае компьютеры способны напрямую взаимодействовать между собой, осуществляя произвольные обмены данными. Во втором – связи между компьютерами клиентами и компьютером-сервером имеют конфигурацию многолучевой звезды.

3.1. ВЫЧИСЛИТЕЛЬНЫЕ ЗАДАЧИ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ, НЕ ТРЕБУЮЩИЕ ПРЯМЫХ ОБМЕНОВ

С точки зрения системных свойств архитектуры «клиент-сервер» следует отнести к самому простому из возможных видов сетевых архитектур. Поэтому с нее началось массовое освоение глобального сетевого пространства. В гипертекстовой сетевой архитектуре «клиент-сервер» построено глобальное информационное пространство *WWW*. Ее функционально более богатые варианты положены в основу распределенных корпоративных систем обработки данных и управления. Данная архитектура широко используется в коммерческих и государственных разработках, таких как построение информационных / управляющих систем в бизнес-приложениях, для медицинских и учебных заведений, в организации научных электронных библиотек, в задачах централизованного сбора и хранения налоговой отчетности и т. п.

В архитектурах этого типа, см. рис. 2, б, обработка может проводиться в двух «противоположных» режимах:

1) задания на обработку готовятся на многочисленных клиентских компьютерах, а обработка запросов и возврат результатов, передаваемых через сеть, осуществляется высокопроизводительными серверными центрами;

2) в серверных компьютерах большая вычислительная задача разбивается на независимые части, рассылаемые многочисленным клиентам-исполнителям, которые по завершению своих заданий возвращают результаты серверным компьютерам.

В первом случае обеспечивается обслуживание большого числа потребителей, во втором – распараллеливание исполнения слабосвязанных фрагментов задач на разных компьютерах.

Для задач с большим объемом вычислений (случай 2) накоплен большой опыт широкомасштабного распараллеливания распределенных вычислений в ресурсах глобальной сети в режиме метакомпьютинга. Существует множество убедительных примеров.

В медицине и биологии:

- *Rosetta@home* – вычисление структуры белка,
- *Folding@Home* – проект по расчету третичной структуры белков,
- *World Community Grid* – изучение белков и их каталогизация,
- *grid.org* – проект по синтезу лекарства от рака,
- *Find-a-Drug* – проект по поиску лекарств от различных болезней.

В естественных науках:

- *SETI@Home* – проект поиска радиосигналов внеземных цивилизаций,
- *Einstein@Home* – проект поиска гравитационных волн,
- *Climate Prediction* – проект по моделированию климата Земли,
- *SpinHenge@home* – проект в области нанотехнологий,
- *LHC@home* – расчеты для создания ускорителя заряженных частиц.

Эта простейшая организация распределенных параллельных вычислений, когда задачи допускают разбиение на слабо-связные фрагменты, исполняемые параллельно, может рассматриваться как демонстрация принципиальной возможности практического применения массовых распределенных вычислений в ресурсах глобальных сетей. На таких задачах количество вовлекаемых компьютеров достигает сотен тысяч и миллионов.

Ограничения, свойственные архитектуре «клиент-сервер»:

- сужение классов решаемых задач распределенной обработки до таких, которые не требуют прямых обменов промежуточными данными между исполняемыми на разных компьютерах фрагментами задач;
- программирование и запуск задач производится только через серверный компьютер, а запуск фрагментов задач на клиентских машинах производится добровольными участниками «вручную».

Ясно, что метакомпьютинг в этой архитектуре нельзя рассматривать как технологию общедоступного программирования и автоматического исполнения распределенных процессов вычислений и управления в ресурсах глобальной сети. Это обстоятельство позволяет сделать вывод, что для эффективного использования функционального и вычислительного потенциала миллиардов компьютерных устройств, связываемых глобальными сетями, требуются принципиально новые сетевые архитектуры распределенных вычислений, формы и представления и способы обработки глобально распределенной информации, методы и средства программирования распределенных вычислений [3-8].

3.2. ЗАДАЧИ С РАВНОПРАВНЫМ РАСПРЕДЕЛЕНИЕМ ФУНКЦИЙ ОБРАБОТКИ ВО ВСЕХ КОМПЬЮТЕРАХ

В общем случае задачи распределенной обработки предполагают наличие сильносвязных фрагментов, исполняемых параллельно и прямые обмены промежуточными данными через сети между вовлеченными компьютерами. Универсально программируемые архитектуры *P2P* в своем предельном потенциа-

ле относятся к функционально общей сетевой организации, которая может стать базовой архитектурой для воплощения универсально программируемых распределенных вычислений и процессов сетевидного управления.

В архитектуре *P2P* функциональные возможности связанных сетями компьютеров должны быть равносильны и в предельном случае могут соответствовать максимальным возможностям универсально программируемых компьютеров. Каждый компьютер должен уметь одновременно действовать как «клиент» и «сервер» в соответствии с функциональными требованиями реализуемого приложения.

В универсально программируемых системах *P2P* компьютеры-участники должны рассматриваться как функционально равноправные объекты, даже если они предоставляют разные объем и качество ресурсов. Важной их чертой является возможность прямого взаимодействия компьютеров между собой, см. рис. 2, а. При этом компьютерная среда оказывается принципиально децентрализованной. Степень децентрализации может варьироваться от частичной до полной. В общем случае выделение фиксированных центральных серверов становится ненужным. Эта сетевая архитектура за счет равноправной параллельной работы всех компьютеров и распараллеливания процессов обмена данными обеспечивает максимальную пропускную способность распределенной обработки.

При этом каждое из компьютерных устройств может становиться как источником/инициатором своих распределенных программ/процессов, так и соисполнителем таковых, инициированных с любых других узлов сети. Это качество необходимо для сетевидных технологий глобального программирования и компьютерной глобализации парадигмы управления на основе математически однородного поля компьютерной информации в модели исчисления древовидных структур [3, 4].

Отметим отсутствие в настоящее время универсальных «бесшовных» средств индустриального программирования в сетевой архитектуре *P2P*. До сих пор распределенные системы в

архитектурах *P2P* используются для создания в основном узко-профильных решений в следующих областях [14]:

- файлообменные системы;
- приложения мгновенного обмена сообщениями;
- организация сообществ и коллаборативных поисковых систем;
- приложения для организации совместной работы;
- управление информацией и знаниями;
- организация эффективного взаимодействия между предприятиями (*b2b*);
- электронные рынки и информационная коммерция;
- распределенные вычисления;
- доставка и распространение контента.

Примером одной из ближайших по времени сфер потенциально крупномасштабного применения являются системы управления потоками товаров по технологиям *RFID* (радиофицированные идентификационные метки). Они обобщают возможности технологий оптических штрих-кодов, заменяя пассивные метки недорогими микропроцессорными устройствами с двусторонним дистанционным доступом по радиоканалам. Интеллектуальные метки *RFID* открывают принципиально новые возможности полномасштабного перехода к эффективному автоматическому управлению перемещаемыми объектами на всех этапах их жизненного цикла [11].

4. Пример больших систем с сетецентрическим управлением

Весьма перспективная область массового применения систем сетецентрического управления формируется в рамках долгосрочной комплексной программы министерства обороны США *Future Combat Systems (FCS)* [18], в которой предлагаются новые принципы и глобальные «стандарты» управления большими системами в военных применениях.

Программа *FCS* определяет архитектуру перспективных глобально распределенных систем сетецентрического (в едином

информационном пространстве компьютерных сетей) управления боевыми единицами, их группами, системами и средствами обеспечения. Она представляет собой многоуровневую систему сетецентрической интеграции стационарных и мобильных объектов разного назначения, оснащенных встроенным компьютерным интеллектом, в едином информационно-функциональном пространстве управления с их взаимодействием в реальном времени.

На рис. 3 проиллюстрированы сферы комплексной применимости сетецентрического управления всеми видами (стационарных и мобильных) средств ведения боевых действий в общем информационно-временном пространстве, охватывающем все виды действий – сухопутных, воздушных, морских, космических. Одним из главных требований становится обеспечение в реальном времени всех участников достоверной и полной информацией соответствующего профиля компетенции, обеспечивающей своевременное достижение поставленных целей в условиях активного противодействия.

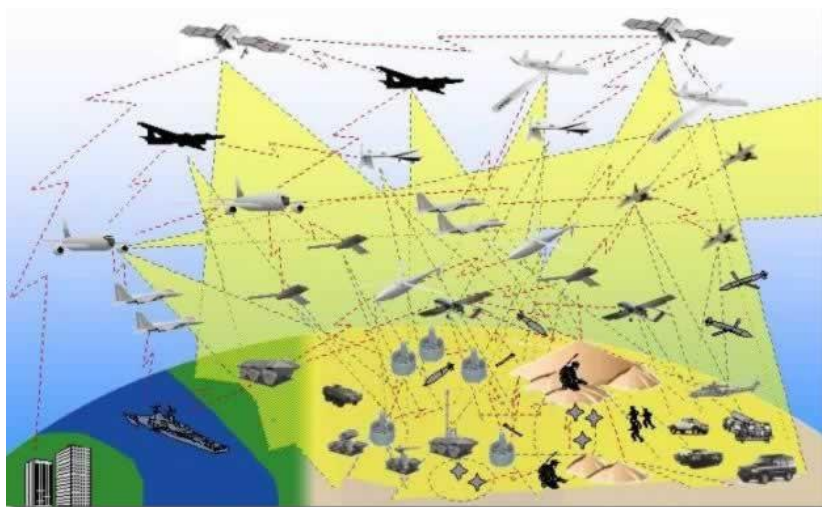


Рис. 3. Сетецентрическое управление «системами систем» мобильных и стационарных объектов [22]

Программа исполняется с начала 2000-х. Практическое использование ее отдельных результатов намечается на 2012-2022 гг. Стратегическая важность программы подтверждается финансированием, которое исчисляется сотнями миллиардов долларов.

Программа *FCS* состоит из 18 компонентов и двух отдельных «надсистем» [2, 13]. Все 18 компонентов разделены на 4 подгруппы: «сухопутные управляемые боевые машины», «сухопутные автоматические боевые машины», «автоматические летательные аппараты», «отдельные устройства».

Ключевым системообразующим элементом программы являются средства связи и компьютерные сети, обеспечивающие поддержание функционально интегрированного информационного поля над зонами боевых действий посредством построения надежной работы мобильных беспроводных сетей. Автономные элементы подключены к общей сети и обмениваются информацией. Так, участники в реальном времени обеспечиваются данными о местоположении противника, поступающими с отдаленных автоматических датчиков, разведывательных машин, беспилотных аппаратов и т.п.

Подобные сетецентрические системы ввиду особой важности имеют высокие приоритеты в финансировании разработок, что позволяет рассматривать их в качестве одной из наиболее вероятных пилотных сфер полномасштабного применения графодинамических систем с управлением.

Сетецентрические системы управления такого рода призваны обеспечить функционирование единого и сверхнадежного информационного пространства, в котором в реальном времени решаются комплексные задачи сбора, накопления и интеллектуальной переработки многоканальных потоков сложно структурированных данных. Цель – формирование единой динамической картины событий и обеспечение превосходящего качества управления большими системами разнопрофильных многокомпонентных систем подвижных и стационарных объектов, гарантирующего достижение поставленных целей с минимальными потерями.

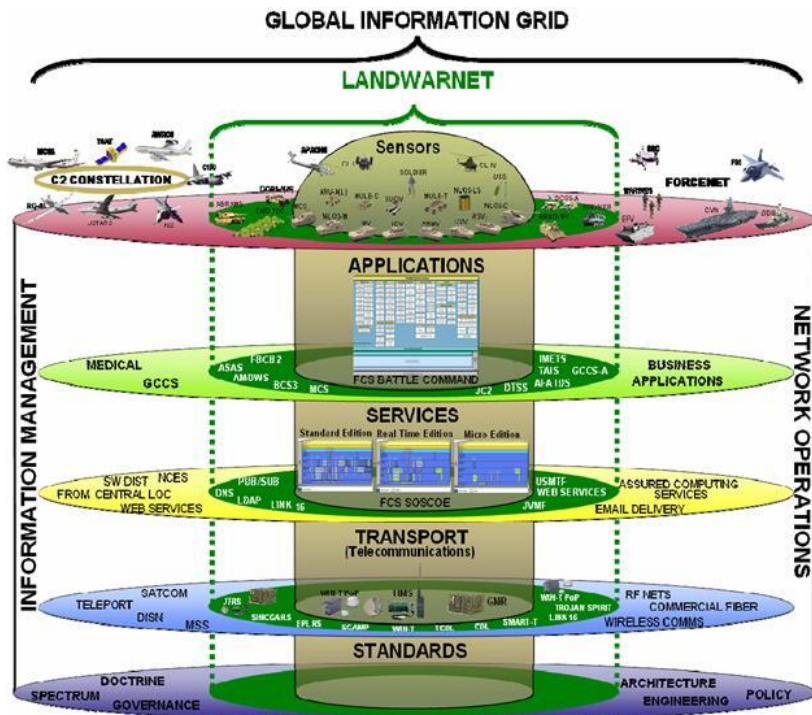


Рис. 4. Многослойная архитектура FCS [23]

В настоящее время концепция FCS, как, впрочем, и все другие сетевые решения, вынуждена опираться на многослойные, разнородные компьютерно-сетевые архитектуры [19] с трудно совместимыми подсистемами различного назначения, реализованных в изначально несовместимых формах представления и способах обработки распределенной компьютерной информации, см. рис. 4.

Отсутствие изначально единого, «бесшовного» универсально-программируемого вычислительного пространства распределенных вычислений и сетечентрического управления в нем, отсутствие адекватной элементной базы и общесистемных решений приводят к комбинаторной сложности интеграции крайне разнородных, с изначально не предусмотренной совмес-

тимостью компьютерных стандартов подсистем и сверхбольшим затратам на их интеграцию.

Затраты в сотни миллиардов долларов до сих пор не привели к созданию единого интегрированного пространства управления (*C4ISR*) для *FCS*. Вместо единого информационно-управляющего пространства, охватывающего все рода войск, создано множество слабо связанных между собой, трудно совместимых узкопрофилированных информационных пространств. Несмотря на технологическое, финансовое и организационное превосходство, задача формирования сильно связанного интегрированного пространства даже в рамках отдельного рода войск остается нерешенной.

Скорее всего, мировой финансово-экономический кризис простимулировал экономически взвешенные взгляды на приоритеты компонентов программы *FCS* [21] и привел к осознанию ограниченных возможностей современных компьютерно-сетевых технологий, которые, несмотря на значительные затраты, в течение 10 лет не смогли дать «лобового» решения центральной задачи *FCS* – формирования единого, системно «бесшовного» информационного пространства сетецентрического управления.

Из сказанного можно сделать вывод о наличии серьезных фундаментальных причин, препятствующих формированию в рамках современных системных качеств (крайне разнородных) компьютерных сред «бесшовного» информационного пространства сетецентрического управления, без которого невозможно полномасштабное решение задач *FCS*.

Отсутствие в руководстве программой *FCS* понимания путей устранения в осязаемой перспективе фундаментальных общесистемных причин, препятствующих формированию единого, функционально полного пространства сетецентрического управления, заставило правительство США с 2009 года пойти на сокращение масштабов задач и перераспределение бюджетных средств программы *FCS* [16, 20].

Несомненно, что доминирование изначальной системной разнородности компьютерно-сетевых решений – наиболее сла-

бое «звено» концепции *FCS*. Десятилетний опыт ее реализации в условиях «обильного» финансирования показывает, что ставка только на подавляющие технологические, финансовые и организационные преимущества не привела к полномасштабному воплощению целей *FCS*.

Для устранения фундаментальных причин изначальной разнородности системных качеств компьютерной среды необходимо формирование в ней единого математически однородного пространства распределенных вычислений и процессов управления, отвечающего следующим требованиям:

- глобальная связность в едином адресном пространстве всех сетевых ресурсов, их универсальная и «бесшовная» программируемость, структурная и системная целостность;
- свободная масштабируемость распределенных данных, программ, процессов вычисления, управления и систем;
- опережающие возможности наращивания структурно-динамической и логической сложности распределенного компьютерного интеллекта;
- независимость системной сложности компьютерной среды от количества компьютеров, связываемых сетями;
- свободно масштабируемая наращиваемость производительности высокопараллельных вычислений в узлах сетей;
- повышенная надежность и устойчивость к деструктивным воздействиям.

Важно отметить, что военные приложения не обладают монополией на сетевые методы управления. Повсеместное проникновение глобальных сетей приводит к тому, что устойчивое развитие мировой социосистемы в целом [4], как чрезвычайно большой и структурно сложной динамической системы, становится невозможным без адекватных, математически корректных моделей сетецентрического управления и их компьютерного воплощения, обеспечивающих согласованное и эффективное функционирование социосистемы и ее частей в условиях глобального информационного пространства. Военные применения, как известно, имеют приоритетное финансирование исследований и разработок, что, несомненно, способствует

полномасштабному практическому продвижению новых методов управления большими системами в глобальном информационном пространстве.

Методы и технологии сетецентрического управления, отработанные на специальных задачах, решаемых в рамках *FCS* и аналогичных программ в других странах, несомненно, дадут импульс к массовому их распространению и полномасштабному решению задач управления посредством глобальной компьютерной среды эффективным функционированием и устойчивым развитием мировой социосистемы в целом и ее частей.

5. Математически однородное поле компьютерной информации на основе исчисления древовидных структур

В настоящее время инструментарий для глобально распределенных вычислений в сетях создается по разнородным, изначально несовместимым моделям и развивается как слабо контролируемая совокупность трудно интегрируемых технологий и поэтому изначально не обладает свойством структурной и системной целостности. Отсюда комбинаторная сложность задач программирования и интеграции распределенных систем, которая выражается чрезмерной трудоемкостью и себестоимостью их разработок.

Это относится и к различным реализациям *Grid*-технологий, которые применяются [10], как правило, для создания систем в рамках проектов ограниченных корпоративными интересами или особо важных применений, для которых аккумулируются требуемые финансовые и интеллектуальные ресурсы.

Одной из главных причин комбинаторной сложности задач интеграции распределенных в сетевых ресурсах данных, программ, процессов и систем является разнородность форм представления и способов обработки компьютерной информации (программ и данных). Избыточное разнообразие форматов компьютерной информации создает в глобальной компьютерной среде эффект непрерывно воспроизводимого «информационно-

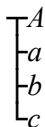
го шума», который принципиально неустраним в рамках известных технологий [3, 4]. Влияние шума быстро нарастает по мере роста масштабов распределенных систем.

Использование единого формализма исчисления древовидных структур позволяет сформировать в сетевых ресурсах математически однородное поле компьютерной информации, в котором доминирующие в настоящее время причины воспроизводства информационного шума (разнородность форм представления данных и программ) устраняются, что открывает возможности построения «бесшовных» технологий программирования распределенных вычислений.

Простейшей процедурной реализацией исчисления древовидных структур является язык и система программирования ПАРСЕК [5], в которой программирование осуществляется посредством универсального базиса операций с древовидными и только древовидными структурами. Они представлены двоичными деревьями и служат для представления как данных, так программ.

4.1. ПЕРВИЧНЫЕ ПОНЯТИЯ ИСЧИСЛЕНИЯ ДРЕВОВИДНЫХ СТРУКТУР [5]

В языке ПАРСЕК для «раскрытия» объекта используется элементарное дерево в геометрической форме:



Дерево состоит из вершин. Вершина характеризуется содержимым вершины и связями с другими вершинами. Имеется 3 вида связей для каждой вершины:

- 1) *deep* – связь с подчиненной по уровню вершиной;
- 2) *next* – связь с нижеследующей вершиной одного уровня;
- 3) *prev* – связь с предыдущей вершиной.

В таблице 2 приведен пример дерева и анализ его структуры. Двоичный набор из признаков наличия или отсутствия каждого из трех видов связей (*prev*, *deep*, *next*) образует статус

вершины. Статусы вершин (связи дерева) отображаются с помощью псевдографических символов \top , \vdash , \perp , а также вспомогательного символа $|$, который дополняется в левой части строк, несущих изображение вершин старшего ранга.

Таблица 2. Определение структуры двоичного дерева

Дерево	Номер вершины	Содержимое вершины	Связь <i>prev</i>	Связь <i>deep</i>	Связь <i>next</i>
	1	<i>OBJ</i>	-	с верш. 2	-
	2	<i>A</i>	с верш. 1	с верш. 3	с верш. 5
	3	<i>a1</i>	с верш. 2	-	с верш. 4
	4	<i>a2</i>	с верш. 3	-	-
	5	<i>B</i>	с верш. 2	-	-

Древовидная структура представляет те или иные вполне структурированные информационные объекты. Обработка объекта ведется последовательно по компонентам. Для выделения компонентов, которыми являются вершины, вводится понятие *курсор*.

Курсор – это переменная, которая принимает значение адреса (ссылки, указателя) некоторой вершины. На дереве можно определить один или более одного курсора. Процесс обработки начинается с открытия курсоров и перемещения их по дереву.

С курсором связан базисный набор функций для работы с деревьями. Они разделяются на следующие группы:

1. *Одношаговое перемещение курсора*

_deep() – переход на подчиненную вершину;

форма вызова: ***cur2 = _deep(cur1)***, где *cur1* – адрес текущей вершины, *cur2* – адрес вершины, подчиненной текущей.

_next() – переход на нижнюю вершину того же ранга;

форма вызова: ***cur2 = _next(cur1)***,

_prev() – переход на предыдущую вершину.

форма вызова: ***cur2 = _prev(cur1)***,

2. *Функции анализа и замены содержимого*

_cont() – прочитать содержимое вершины;

форма вызова: ***c = _cont(cur)*** или ***c = {cur}***,

где *s* – содержимое вершины, *cur* – адрес вершины (курсор).

_re_cont() – занесение нового содержимого вершины.

форма вызова: **_re_cont(*cur*, NC)** или **{*cur*} = NC**,

где *cur* – адрес вершины, NC – новое содержимое.

3. Создание и уничтожение вершин

_node_create() – создать вершину,

форма вызова: ***cur* = _node_create(NC)**,

где *cur* – переменная (с произвольным именем), которая является вновь создаваемым курсором, указывающим на созданную вершину, NC – новое содержимое. Каждая создаваемая вершина является изолированной.

_node_del() – удалить вершину,

форма вызова: **_node_del(*cur*)**,

где *cur* – текущий курсор.

4. Редактирование связей

_link_next(,) – связать две вершины на одном уровне;

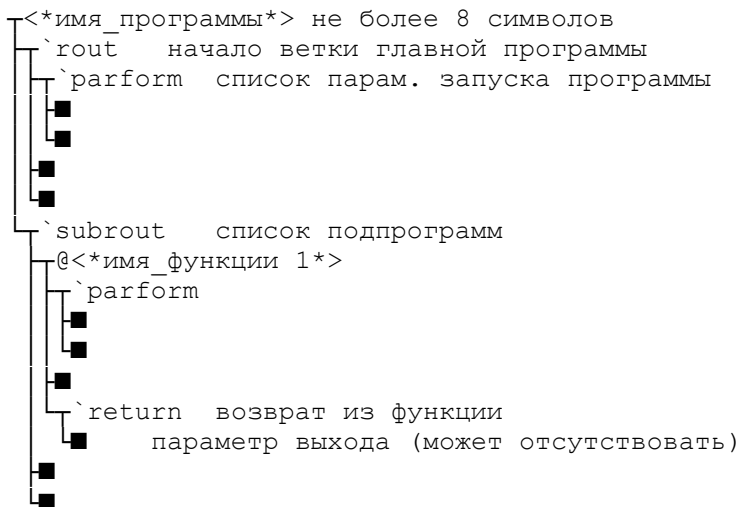
_link_deep(,) – связать две вершины так, чтобы вторая стала подчиненной первой;

_link_del(,) – разорвать связь между вершинами,

формы вызова: **_link_next(*cur1*, *cur2*)**, **_link_deep(*cur1*, *cur2*)**, **_link_del(*cur1*, *cur2*)**, где *cur1* – адрес первой вершины, *cur2* – адрес второй вершины.

4.2. СТРУКТУРА ПАРСЕК-ПРОГРАММЫ [5]

Структура ПАРСЕК-программ вполне классическая и типична для процедурного программирования. Особенность лишь в том, что тексты программ представляются в виде двоичного дерева с ключевыми словами:



Символьные строки в правой части вершины дерева после первого пробела образуют комментарий (например: строка «начало ветки главной программы»).

Условные обозначения:

1. █ – поддереву или лист;
2. Ключевые слова в языке ПАРСЕК начинаются с символа обратный апостроф «`», т. е.

`<*ключевое_слово*> == `<*идентификатор*>.`

С помощью ключевых слов строятся все конструкции языка.

3. Идентификатор встроенных функций начинается с символа «_», пользовательских – с «@», т. е.

`<*встроенная_функция*> == _<*идентификатор*>.`

`<*пользовательская_функция*> ==`

`@<*идентификатор*>.`

Базовые конструкции языка ПАРСЕК вполне традиционны для процедурного программирования:

1. Циклы

```
| `repeat          <<- цикл с постусловием  
| | <*тело*>  
| | `until  
| | | <*условие*>
```

```
| `while          <<- цикл с предусловием  
| | <*условие*>  
| | `do  
| | | <*тело*>
```

2. Условия

```
| `if            <<- условный оператор  
| | <*условие*>  
| | | `then  
| | | | <тело>  
| | | `else <<- else - необязательный параметр  
| | | | <иначе>
```

```
| `case         <<- многовариантный выбор  
| | <*условие*>  
| | | `of  
| | | | <*константа 1*>  
| | | | | <*тело 1*>  
| | | | <*константа 2*>  
| | | | | <*тело 2*>  
| | | | ...  
| | | | <*константа N*>  
| | | | | <*тело N*>  
| | | `default  
| | | | <*иначе*>
```

Операция присваивания, выражения и функции в древовидной записи представлены в таблице 3. Благодаря уникальным свойствам деревьев (удаление любого ребра нарушает связность, число ребер на 1 меньше числа вершин) математически однородное поле компьютерной информации в модели исчисления древовидных структур обладает неулучшаемым свойством минимальной структурной сложности.

Таблица 3. Представление операций и выражений

Конструкция	Формульный вид	Древовидная запись
присваивание	$a = b$	$\begin{array}{c} \top a \\ \mid \\ \perp b \end{array}$
операция	$1 + 3$	$\begin{array}{c} \top + \\ \mid \\ \perp \begin{array}{c} 1 \\ 3 \end{array} \end{array}$
функция	$\begin{array}{l} _f1(x, y) \\ @f2(x, y) \end{array}$	$\begin{array}{cc} \top _f1 & \top @f2 \\ \mid & \mid \\ \perp \begin{array}{c} x \\ y \end{array} & \perp \begin{array}{c} x \\ y \end{array} \end{array}$
выражение	$a = 1 + 3$	$\begin{array}{c} \top a \\ \mid \\ \perp \begin{array}{c} + \\ \perp \begin{array}{c} 1 \\ 3 \end{array} \end{array} \end{array}$

Математическая замкнутость исчисления древовидных структур позволила построить простые и эффективные алгоритмы автоматического управления процессами исполнения программ как во внутренней памяти компьютеров, так и в едином адресном пространстве оперативной памяти компьютеров, связанных сетями.

4.3. РАСПРОСТРАНЕНИЕ ИСЧИСЛЕНИЯ ДРЕВОВИДНЫХ СТРУКТУР НА РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ

Принципы распространения формализма исчисления древовидных структур на распределенные вычислительные сетевые ресурсы изложены в [3, 4, 6, 7]. При этом достигается распространение свойства универсальной программируемости с внутренних ресурсов компьютеров на распределенные сетевые ресурсы. Для этого с использованием базовых функций управления протоколом *TCP/IP* обеспечивается формирование единого адресного пространства распределенной памяти всех связанных сетями компьютеров, участвующих в распределенной обработке.

Единое адресное пространство, охватывающее память всех доступных компьютеров, позволяет программисту «бесшовно»,

в едином логическом базисе, работать с древовидными структурами, размещаемыми как в памяти собственного компьютера, так и в памяти отдаленных компьютеров, см. рис. 5.

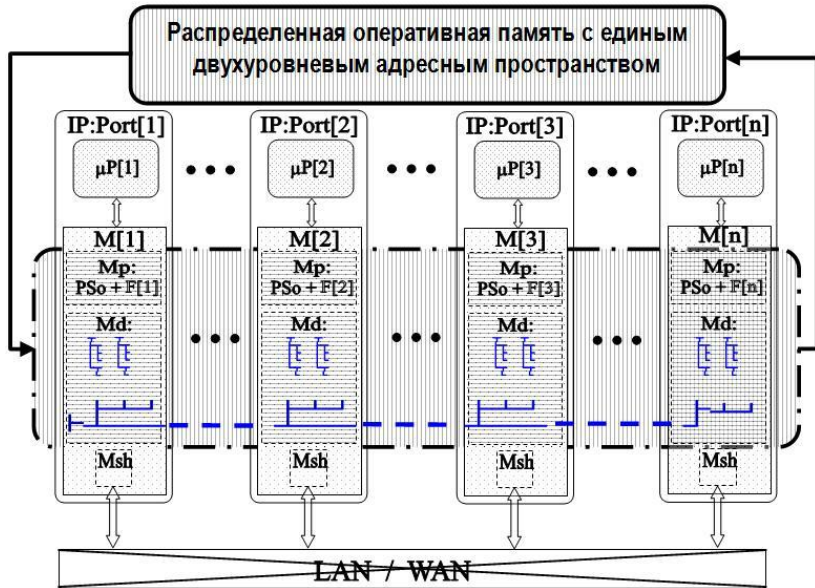


Рис. 5. Виртуальная Парсек-машина для распределенных вычислений

Единое («бесшовное») пространство оперативной памяти компьютеров, связанных сетями состоит из двухкомпонентных адресов. Первая компонента – уникальный сетевой IP -адрес компьютеров, вторая – адреса ячеек их оперативной памяти [6, 7].

Распространение свойства универсальной программируемости с внутрикомпьютерных ресурсов на распределенные сетевые ресурсы предполагает следующее. Все ячейки оперативной памяти отдаленных компьютеров, охваченные единым адресным пространством, становятся доступными с каждого из компьютеров («бесшовность»). Различные данные, обрабатываемые программой, целиком или своими компонентами могут распределяться по различным компьютерам таким образом, что

адресный доступ к ним из программы может осуществляться «бесшовно», т. е. без применения дополнительных языковых средств. Например, в выражении $C := A + B$ все три операнда могут находиться в ОЗУ трех разных компьютеров, операция будет выполнена так, как если бы все они находились в памяти одного компьютера, т.е. с точки зрения программиста сеть полностью прозрачна. Такое свойство «бесшовности» относится не только к арифметическим, логическим или символьным выражениям, но и к командам условного перехода, циклам, вызовам подпрограмм (с передачей фактических параметров, находящихся в разных компьютерах), к операторам ввода/вывода, позволяющим читать/записывать файлы, расположенные на разных компьютерах. Это касается всех конструкций языка программирования ПАРСЕК.

Единое пространство двухкомпонентных адресов [6] введено в систему программирования ПАРСЕК [5, 7]. Для организации распределенных вычислений, обеспечения их координированного взаимодействия и синхронизации вычислительных процессов в язык ПАРСЕК введены функции [6, 7, 8], с помощью которых задается распределенная обработка компонентов деревьев посредством одновременного исполнения на разных компьютерах многих асинхронно взаимодействующих процессов. Основными компонентами распределенных вычислений являются как перемещаемые *exe*-модули, так и подпрограммы (функции) с параметрами отдаленного запуска и согласованными интерфейсами межкомпонентного сетевого взаимодействия по событиям и данным.

Расширение сфер программирования на сетевые ресурсы осуществляется посредством встраивания в систему ПАРСЕК механизмов отдаленного запуска процедур *RPC* (*Remote Procedure Calls*). Встраиваемые механизмы логически и технологически прозрачны для программистов и не требуют сложного системно-сетевого конфигурирования и сопровождения. Это открывает перспективы общедоступного программирования распределенных структурно сложных вычислений в сколь угодно большой совокупности связанных компьютеров.

Как известно, концепция *RPC* расширила сферы программирования с внутренних ресурсов компьютеров на распределенные вычислительные ресурсы сетей [15, 17]. Механизмы *RPC* в реализации существенно сложнее и медленнее вызова локальных процедур, т. к. задействуют инерционные системно-сетевые механизмы. Существенный недостаток известных реализаций *RPC* состоит в том, что взаимодействие процессов, протекающих в оперативной памяти на отдаленных компьютерах, осуществляется в различных адресных пространствах. Поэтому в принципе отсутствуют возможности работы с глобальными переменными и передачи адресных указателей в качестве параметров процедур. Неустраненные логические и технологические различия в вызове локальных и отдаленных процедур препятствуют их идентичности, что привносит дополнительные проблемы программирования, которые трудно решаются в условиях разработки современных распределенных систем, которые должны гибко и оперативно реагировать на структурные изменения информационного контекста глобально-информационного пространства.

Причины таких ограничений не в существующих способах реализации *RPC*, а в принципиальных ограничениях сфер действия классической компьютерной аксиоматики, которая замыкает свойство универсальной программируемости на внутренних ресурсах изолированных компьютеров. В таких ограничениях адресное пространство изначально локализовано ячейками внутренней памяти. Поэтому в существующих языках и системах программирования, наследующих это ограничение, простые и структурированные данные, способы их представления в памяти и обработки изначально ориентированы на размещение в локальной памяти. Механизмы *RPC* в отсутствие универсальной структурно целостной и функционально полной модели распределенных вычислений вынужденно реализуются на уровне частных способов стыковки разнородных адресных пространств, что является существенным препятствием на пути к технологиям «бесшовного» программирования распределенных вычислений в сетевых ресурсах.

Для распространения свойств универсальной программируемости системы ПАРСЕК с внутренних ресурсов компьютеров на распределенные вычислительные ресурсы локальных и глобальных сетей с использованием библиотеки функций управления протоколом *TCP/IP* реализован базис управления распределенными вычислениями в сетевой архитектуре *Peer-to-Peer*.

Для присоединения компьютеров к распределенной системе требуется установка небольшой программы поддержки сетевого взаимодействия вовлекаемых компьютеров, которая передается при регистрации их *IP*-адресов.

Система программирования ПАРСЕК, расширенная функциями распределенных вычислений в едином адресном пространстве исчисления древовидных структур, испытана на задаче поиска оптимальных конфигураций коммутирующих сетей на основе комбинаторных методов построения квазиполных графов (симметричных блок-схем) [9]. Алгоритм решения задачи допускает разбиение на многие слабосвязанные фрагменты. Это позволяет исполнять их одновременно на многих компьютерах связанных через Интернет. Обеспечивается сокращение времени счета, пропорциональное числу вовлеченных компьютеров. В ходе экспериментов в распределенных вычисления участвовало до 20 компьютеров различных локальных сетей нескольких организаций, взаимодействующих между собой через Интернет.

6. Переход в единое функциональное пространство древовидных структур

Предлагаемый на основе исчисления древовидных структур подход открывает возможности придания глобальной сети свойства универсальной программируемости и системной нацеленности на решение разнообразных задач управления в графодинамических системах с сетецентрическим управлением.

Важно отметить, что в процедурном языке ПАРСЕК работа с древовидными структурами, так же как в функциональном языке ЛИСП, не сужает сфер их универсальной применимости.

Сведение общего случая структурно сложных задач к древовидным структурам с процедурным стилем программирования позволило существенно упростить компьютерные реализации системы программирования ПАРСЕК [5] и, одновременно, повысить их эффективность.

Универсальный язык ПАРСЕК и его система программирования открывают возможности программирования графодинамических систем с сетцентрическим управлением в математически однородном поле компьютерной информации, охватывающем ресурсы глобальной компьютерной среды. В связи с этим ориентация графодинамики в своем общем случае на графы общего вида [1] представляется излишней. Произвольные преобразования сложно структурированной информации можно обеспечивать с помощью деревьев – самых простых связных графов, имеющих минимальную структурную сложность (число ребер на единицу меньше числа вершин). Ограничение графодинамики древовидными структурами является существенным фактором снижения трудоемкости программирования графодинамических систем в ресурсах глобальных компьютерных сетей.

Отметим, что представление деревьев с помощью нумерующих функций, используемое в публикации [1], имеет два принципиальных недостатка:

- чрезвычайно затруднена смысловая интерпретация математических преобразований нумерующих функций, представляющих деревья;
- компьютерная реализация деревьев с помощью нумерующих функций ведет к крайне неэффективному использованию оперативной памяти и процессорного времени.

В компьютерном исчислении древовидных структур системы программирования ПАРСЕК данные недостатки устранены. Эти качества, а также распространение свойства универсальной программируемости распределенных (в архитектуре *P2P*) структурно сложных вычислений на любые совокупности связанных сетями компьютеров позволяют рассматривать ПАРСЕК как основу для построения новых промышленных средств про-

граммирования, ориентированных на создание графодинамических систем с сетецентрическим управлением.

Проблемы компьютерно-сетевого управления большими структурно-сложными распределенными системами до настоящего времени имеют три разноплановых (разнородных) уровня рассмотрения: математические модели управления (МС), системно-программная среда *SoftWare* (SW), машинная среда *HardWare* (HW).

На рис. 6 показан переход от трех, ныне неизбежных, изначально разнородных уровней рассмотрения больших систем в различных понятийных пространствах (слева) к однородному и математически замкнутому пространству сетецентрического управления (справа).

В новом пространстве чрезвычайно сложные и разноплановые, но сугубо внутренние, компьютерные проблемы машинного и системно-программного уровней исключаются из круга содержательных проблем управления большими системами. При этом открываются возможности воплощения математических моделей управления в сколь угодно большой компьютерной среде в едином математически замкнутом компьютерном формализме исчисления древовидных структур, реализованном в языке и системе программирования ПАРСЕК. Этот подход освобождает специалистов в области управления от погружения в специфические системно-программные проблемы (многоаспектные и разнородные, потому чрезвычайно трудоемкие и трудно контролируемые) глобальной машинной среды.

При доминирующей в настоящее время изначально разнородности всех трех уровней функционирования компьютерной среды с увеличением ее размера с каждым добавляемым компьютером привносятся открытые степени свободы, которыми необходимо управлять. Известно, что общесистемная сложность с линейным ростом размера и привносимых степеней свободы увеличивается сверхлинейно (комбинаторно). Привносимые степени свободы управления компьютерной среды открыты на человека. Именно поэтому ее устойчивое функционирование до сих пор невозможно без непрерывного сопровождения огром-

ной армией программистов, системных и сетевых администраторов.

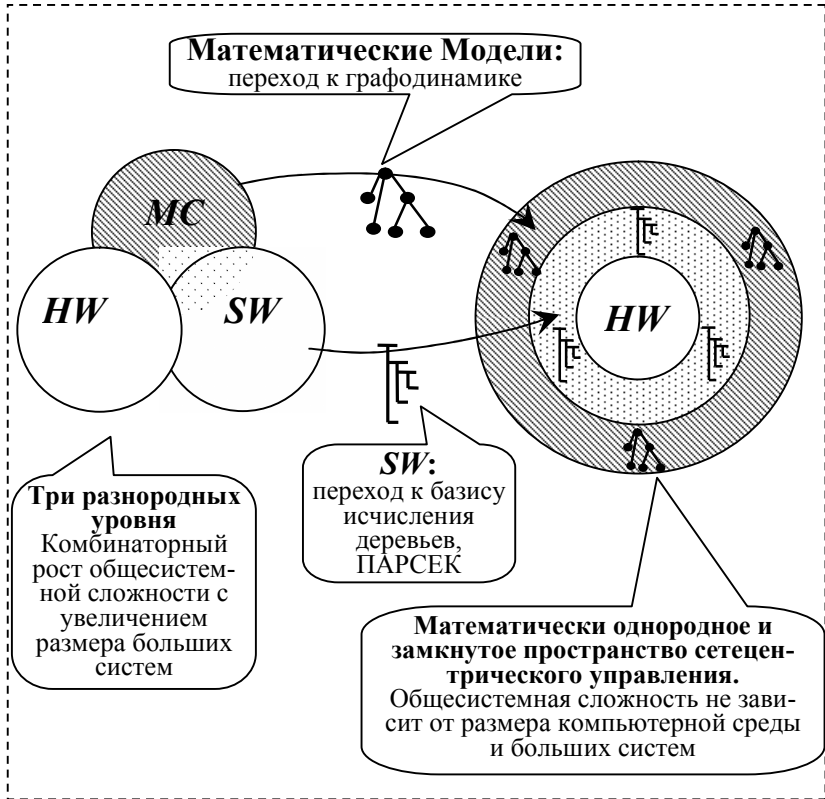


Рис. 6. Переход в математически однородное и замкнутое пространство сетецентрического управления в едином формализме исчисления древовидных структур

В математически однородном поле компьютерной информации на основе исчисления древовидных структур собственные системные степени свободы глобальной машинной среды «скрываются» под внешними математическими оболочками

единого для графодинамики и системы программирования ПАРСЕК математически замкнутого формализма исчисления древовидных структур (см. рис. 6, справа, два внешних кольца). В этом случае *общесистемная сложность компьютерной среды перестает зависеть от своего размера*, а собственные внутри-системные проблемы управления степенями свободы глобальной компьютерной среды могут быть, в принципе, выведены из сфер прямого участия человеческого фактора. В такой ситуации содержательные проблемы создания больших систем с сетецентрическим управлением разнообразного потребительского назначения не подменяются чрезвычайно сложными, трудоемкими и трудно контролируруемыми проблемами управления машинной средой.

Графодинамические модели (см. рис. 1) больших систем имеют в представлениях с дискретным временем привычный вид системы уравнений в векторной форме:

$(y[t], q[t]) = F(q[t-1], x[t], u[t]), t = 1, 2, \dots;$ где $x[t], y[t], q[t], u[t]$ – последовательности значений входных, выходных векторов, векторов внутренних состояний и управляющих воздействий, соответственно:

$x[t] = (x_1[t], x_2[t], \dots, x_n[t]) \in X, x_i[t] \in X_i, i = 1, 2, \dots, n;$

$y[t] = (y_1[t], y_2[t], \dots, y_k[t]) \in Y, y_i[t] \in Y_i, i = 1, 2, \dots, k;$

$q[t] = (q_1[t], q_2[t], \dots, q_m[t]) \in Q, q_i[t] \in Q_i, i = 1, 2, \dots, m;$

$u[t] = (u_1[t], u_2[t], \dots, u_r[t]) \in U, u_i[t] \in U_i, i = 1, 2, \dots, r;$

$F(, ,)$ – вектор-функция:

$F(, ,) = (f_1(, ,), f_2(, ,), \dots, f_p(, ,)) \in F, f_i \in F_i, i = 1, 2, \dots, p.$

Особенность данного представления состоит только в том, что для всех переменных области их определения X_i, Y_i, Q_i, U_i вложены в множество всех деревьев $D: X_i, Y_i, Q_i, U_i \subseteq D$, а все функции $f_i(, ,)$ строятся в базисе исчисления древовидных структур.

Отметим принципиальное различие методологических целей графодинамики и системы программирования в сквозном формализме исчисления древовидных структур.

Система программирования основывается на исчислении древовидных структур с *универсальным базисом простейших*

операций над деревьями, обладающим свойством функционально полноты: любое дерево может быть построено композициями простейших операций над другими деревьями. В системе ПАРСЕК [5] такие композиции реализуются при программировании посредством универсального набора структур управления, используемого во всех процедурных языках (линейный участок, ветвления, циклы, вызовы подпрограмм).

Цель графодинамических методов – классификация структурно и динамически сложных систем с управлением, построение и исследование математических моделей управления. Определение разнообразия классов динамических систем опирается на различные свойства математической симметрии математических объектов, из которых строятся модели. Различные свойства симметрии, которые определяют разные классы графодинамических систем, предполагают построение соответствующих базисов работы с деревьями более высокого логического уровня, нежели тот, который лежит в основе универсальной системы программирования. Базисы высокого уровня должны отражать свои для разных классов графодинамических систем математические симметрии.

Главные методологические цели теоретических исследований графодинамических систем – разработка и обоснование математических базисов высокого уровня, построение (анализ и синтез) моделей управления в эти базисах, исследование свойств их корректности и эффективности.

Главные методологические цели программной реализации разработанных математических моделей – программирование базисов высокого уровня (создание библиотек таких базисов), программирование в соответствующих библиотечных базисах высокого уровня разработанных теоретиками математических моделей управления.

7. Выводы

В условиях экспоненциального роста сфер влияния глобальной компьютерной среды графодинамические системы с

сетецентрическим управлением следует рассматривать как качественно новый системообразующий фактор преодоления растущей структурной и динамической сложности больших и сверхбольших распределенных систем управления, реализуемых в глобальном информационном пространстве компьютерной среды.

Такие графодинамические системы являются носителями и преобразователями структурно сложной информации в технических, организационных, социальных и других применениях. Однако технологии индустриального программирования пока не предоставляют общедоступных средств их построения. Причина в том, что они изначально не обладают способностью распространять свойство универсальной программируемости на совокупные ресурсы любых компьютеров, связанных сетями. Это свойство достижимо в математически однородном поле компьютерной информации, основанном на исчислениях древовидных структур.

Приходит время, когда глобальную компьютерную среду в целом необходимо рассматривать как принципиально новую, универсальную по исполняемым функциям, техническую систему, все компоненты которой изначально обладают свойствами сильной связности. Эта система сможет взять на себя роль управляющего универсума – носителя потенциально неограниченного множества систем и процессов управления высокой структурно-динамической сложности. Конкретные сетецентрические системы управления в такой среде будут собираться выбором сильно связанных компонентов (из множества доступных сетевых ресурсов) с требуемыми свойствами и функциями и программированием только необходимых связей между ними и недостающих функций.

Литература

1. АЙЗЕРМАН М.А., ГУСЕВ Л.А., СМИРНОВА И.М., ПЕТРОВ С.В. *Динамический подход к анализу структур*,

- описываемых графами (Основы графодинамики) // *АиТ*. – 1977. – №7. – С. 135-151. – №9. – С. 123–136.
2. ВЭЙ С. *Американские перспективные боевые системы*. URL: <http://www.army-guide.com/rus/article/article.php?forumID=220> (дата обращения: 23.10.2010).
 3. ЗАТУЛИВЕТЕР Ю.С. *На пути к глобальному программированию* // *Открытые системы*. – 2003. – №3. – С. 46–47. – URL: <http://www.osp.ru/os/2003/03/182704/> (дата обращения: 23.10.2010).
 4. ЗАТУЛИВЕТЕР Ю.С. *Проблемы глобализации парадигмы управления в математически однородном поле компьютерной информации* // *Проблемы управления*. – 2005. – №1. – Ч.1. – С. 1-12. – №2. – Ч.2. – С. 13–23. – URL: <http://zvt.hotbox.ru> (дата обращения: 23.10.2010).
 5. ЗАТУЛИВЕТЕР Ю.С., ХАЛАТЯН Т.Г. *ПАРСЕК – язык компьютерного исчисления древовидных структур с открытой интерпретацией. Стеновый вариант системы программирования*. – М.: Институт проблем управления РАН, 1997. – 71 с.
 6. ЗАТУЛИВЕТЕР Ю.С. *Компьютерная информация в модели исчисления древовидных структур* / Труды Второй международной конференции «Идентификация систем и задачи управления». – М.: Институт проблем управления им. В.А. Трапезникова РАН, 2003. – С. 790–858.
 7. ЗАТУЛИВЕТЕР Ю.С., ТОПОРИЦЕВ А.В. *Язык Парсек: программирование глобально распределенных вычислений в модели исчисления древовидных структур* // *Проблемы управления*. – 2005. – №4. – С. 12–20.
 8. ЗАТУЛИВЕТЕР Ю.С., ФИЩЕНКО Е.А. *Организация распределенных вычислений в системе программирования ПАРСЕК на примере сжатия цифрового видео*. // *Проблемы управления*. – 2003. – №4. – С. 6–10.
 9. КАРАВАЙ М.Ф., ПАРХОМЕНКО П.П., ПОДЛАЗОВ В.С. *Комбинаторные методы построения двудольных однородных минимальных квазиполных графов (симметричных блок-схем)* // *АиТ*. – 2009. – №.2. – С. 153–170.

10. КОВАЛЕНКО В.Е., КОРЯГИН Д. *Организация grid: есть ли альтернативы?* // Открытые системы. – 2004. – №12. – С. 34–41.
11. КРОТОВ В. *Интернет вещей* // Computerworld. – 2008. – №3 – URL: <http://www.osp.ru/cw/2008/03/4755084> (дата обращения: 23.10.2010).
12. КУЗЬМИН И. *Future Combat System – революция или эволюция?* – URL: http://www.3dnews.ru/editorial/future_combat_system (дата обращения: 23.10.2010).
13. НИКОЛАЕВ А. *Боевые системы будущего.* URL: http://www.militaryparitet.com/html/data/ic_news/227/ (дата обращения: 23.10.2010).
14. СУХОРОСЛОВ О.В. *Пиринговые информационные системы: концепция, анализ и приложения.* – М: МФТИ, 2002. – 106 с.
15. ТАНЕНБАУМ Э., ВАН СТЕЕН М. *Распределенные системы. Принципы и парадигмы.* – Спб.: Питер. – 2003. – 877 с.
16. ШАДРИН И. *Программа модернизации американской армии Future Combat Systems подвергнется серьезной реорганизации.* URL: http://www.infox.ru/hi-tech/tech/2009/04/08/FCS_fail.phtml . (дата обращения: 23.10.2010).
17. BIRREL A.D., NELSON B.J. *Implementing Remote Procedure Calls* // ACM Trans. Comp. Systems. – 1984. –Vol.2, №1. – P. 39–59.
18. *FCS 2005 Flipbook Boeing*, URL: <http://www.globalsecurity.org/military/library/report/2005/050000-fcs2005flipbook.pdf> (дата обращения: 23.10.2010).
19. *FCS Network*. URL: <http://www.globalsecurity.org/military/systems/ground/fcs-network.htm> (дата обращения: 23.10.2010).
20. GATES R.M. *Defense Budget Recommendation Statement* URL: <http://www.defense.gov/speeches/speech.aspx?speechid=1341> (дата обращения: 23.10.2010).

21. *The Army's Future Combat Systems' Features, Risks, and Alternatives*. URL: <http://www.gao.gov/new.items/d04635t.pdf> (дата обращения: 23.10.2010).
22. URL: <http://www.darkgovernment.com/news/darpa-looks-for-ultimate-tracking-ability/> (дата обращения 23.10.10)
23. URL: http://en.wikipedia.org/wiki/FCS_Network (дата обращения: 23.10.2010).

GRAPH-DYNAMIC SYSTEMS WITH NETWORK-CENTRIC CONTROL IN MATHEMATICALLY UNIFORM FIELD OF COMPUTER INFORMATION

Yurii Zatuliveter, Institute of Control Sciences of RAS, Moscow, Cand.Sc., assistant professor (zvi@ipu.rssi.ru, Moscow, Profsoyuznaya st., 65, (495)334-92-09).

Elena Fischenko, Institute of Control Sciences of RAS, Cand.Sc., Moscow, (fish@ipu.rssi.ru, Moscow, Profsoyuznaya st., 65, (495)334-92-09).

Abstract: As development of ideas of the graph-dynamics, the problems are discussed of implementation of systems of high structural complexity with network-centric control in a global computer environment. The approach is proposed to formation of mathematically uniform field of computer information in global networks on the basis of the calculus of tree-like structures. The universal formalism to presentation and "seamless" programming of distributed graph-dynamic control systems is offered as a part of the approach. The examples are given of systems with network-centric control that can be qualified as graph-dynamic systems.

Keywords: computer networks, network-centric control, tree-like structures, graph-dynamics, uniform address space, integration.

*Статья представлена к публикации
членом редакционной коллегии Д. А. Новиковым*