

О ПОДХОДАХ К МОДЕЛИРОВАНИЮ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР КОМПЬЮТЕРНЫХ СЕТЕЙ

В связи с бурным ростом технологий компьютерных сетей, наблюдаемым в последнее время, совершенно закономерным является повышенное внимание к проблеме проектирования корпоративных информационно-телекоммуникационных систем (КИТС), являющихся подмножеством распределенных информационно-вычислительных систем (РИВС). Это объясняется тем, что по мере увеличения масштаба компьютерных сетей и ужесточения требований, предъявляемых к создаваемым системам в плане автоматизации деятельности предприятий, интуитивный поиск решения и структуры проекта разработчиком неприемлем, и не может привести к удовлетворительным результатам. Решение подобных задач требует использования систем автоматизации проектирования (САПР), а, следовательно, формализации процесса проектирования на достаточно детальном уровне в рамках какой-либо модели, позволяющей на основе ее «прогона» и анализа полученных результатов произвести синтез структуры программного комплекса.

К решению данной проблемы существует несколько альтернативных подходов.

Развиваемый в работах Э. В. Зиновьева, В. В. Кульбы, Н. И. Листопада, К. С. Тониева, А. Д. Цвиркуна и других подход (подход 1) основывается на понимании сети как связного графа, вершины которого представляют рабочие станции и элементы сетевой коммуникации [1, 2]. При этом, программный комплекс представляется набором узлов сети, выполняющих вычислительные работы и обменивающихся данными. В такой постановке задача проектирования РИВС сводится, в общем случае, к размещению информационных массивов баз данных по узлам сети, а также закреплению вычислительных работ за конкретными узлами.

Сильной стороной данного подхода является относительная простота получаемых моделей функционирования системы, и, как следствие, вычислительная простота подлежащих минимизации (с целью оптимизации структуры системы) функционалов. Это позволяет использовать хорошо исследованные методы целочисленного программирования для решения задачи на ЭВМ.

В качестве недостатка, как следует из самой постановки задачи, можно отметить высокий порядок абстракции при построении модели сетевого взаимодействия, вынуждающий отказаться от рассмотрения особенностей, специфичных для тех или иных программных комплексов. Не

рассматриваются задержки в передаче данных по сети, обусловленные состязательным методом доступа к каналу передачи, не рассматривается влияние репликации и синхронизации данных на работу системы и т. д.

Это приводит к тому, что полученные результаты в достаточно малой степени соответствуют действительности, и их практическая ценность невелика. Наибольшей популярностью данный подход пользовался в 1980–1990 гг., в настоящее же время он используется редко. Его применение приемлемо для проектирования распределенных приложений, используемых в территориально разобобщенных сетях со слабым взаимодействием компонентов неопределенного типа, редким изменением присутствующих в системе данных, иррелевантностью по отношению к ходу вычислительного процесса. Для проектирования современных корпоративных программных комплексов в локальных сетях его использование видится нецелесообразным.

Рассматриваемый в работах И. В. Агеенко, М. А. Вагих, А. В. Воружева, О. М. Демиденко, И. В. Максимеева, А. Б. Демуськова, О. И. Еськовой и других подход (подход 2) основывается на понимании вычислительной сети ЭВМ как набора систем массового обслуживания [3]. Далее осуществляется декомпозиция каждой ЭВМ на структурные блоки, такие как память, жесткий диск, процессор, с присущими каждому блоку характеристиками. Рассматриваются три типа присутствующих в системе процессов: интерактивные, фоновые и транзитные. Далее, с учетом доступных ресурсов системы и присутствующих требований со стороны процессов создаются модели временных задержек в подобной вычислительной системе. В такой постановке задачи с помощью аппарата теории массового обслуживания можно исследовать временные характеристики программного комплекса и на основе этого осуществлять синтез оптимальных структур размещения и взаимодействия его компонентов.

Сильной стороной данного подхода является возможность анализа временных характеристик системы и детализированное рассмотрение процесса функционирования за счет декомпозиции системы на ряд структурных блоков (память, процессор и т. д.), что наряду с возможностью использования при построении модели реальных данных, полученных при мониторинге, позволяет достаточно точно аппроксимировать с помощью модели реальные РИВС.

К недостаткам подхода можно отнести высокую вычислительную сложность решения поставленной задачи, не позволяющую использовать ее при просчете крупных систем. Необходимость в наличии статистических данных, специфических для планируемого типа программного

обеспечения, и больший акцент на процессы мониторинга позволяют говорить о том, что данный подход более приемлем для реинжиниринга существующей КИТС, нежели для синтеза структуры «с нуля».

Подход, применяемый в работах И. И. Румянцевой, К. В. Святенко, В. О. Сиротюка и других (подход 3), является, по сути, синтезом двух рассмотренных выше [4]. Обобщая проводимое ими проектирование такой РИВС, как системы управления распределенными базами данных (СУРБД), можно сформулировать наиболее перспективный, с нашей точки зрения, подход к пониманию вычислительного процесса в сети. Сама сеть понимается в рамках подхода 1, как множество рабочих станций и элементов сетевой коммуникации, объединенных друг с другом каналами передачи данных. Их взаимодействие, однако, записывается в несколько более сложной форме, на более детальном уровне (стремление к детализации, как в подходе 2).

Сильной стороной данного подхода является возможность использования методов оптимизации и операционного исчисления для решения сформулированных задач. Подобная постановка задачи позволяет более точно и адекватно записать схемы сетевого взаимодействия.

В качестве недостатков подхода можно отметить более высокую (по отношению к 1-му подходу) сложность получаемых моделей и необходимость «настройки» на тип решаемой задачи (распределенные базы данных, видеоконференции, средства middleware и проч.). Результаты сравнения подходов между собой представлены в табл. 1.

Из приведенного сравнения видно, что в настоящее время для построения КИТС среднего масштаба в вычислительных сетях наиболее перспективным оказывается подход 3.

Задача построения информационно-вычислительной карты сети является многокритериальной оптимизационной задачей. Для ее решения классически применяются методы случайного поиска (МСП) и методы

Таблица 1

Сравнение подходов к моделированию сетевого взаимодействия

	Подход 1	Подход 2	Подход 3
Размер сети	Средние	Малые	Средние
Узел – структура или точка?	Точка	Структура	Точка
Схема взаимодействия процессов	Задана в виде списка взаимодействия узлов	Задана в виде списка взаимодействия узлов	Формализована, определяется в процессе
Адекватность реальности	Низкая/Средняя	Высокая	Средняя/Высокая
Вычислительная сложность	Низкая/Средняя	Высокая	Средняя/Высокая

градиентного спуска (МГС), позволяющие решать только однокритериальные оптимизационные задачи. Они осуществляют свертку нескольких критериев в один с помощью весовых коэффициентов или трансформируют их в ограничивающие условия. Поэтому сегодня в подобных случаях применяются также методы эволюционных вычислений, а именно генетические алгоритмы (ГА).

ГА – это адаптивные методы поиска, основанные на принципах эволюции и естественного отбора, позаимствованных из теории Дарвина. В их основе лежит утверждение теории эволюции о том, что каждый биологический вид целенаправленно развивается и изменяется для того, чтобы наилучшим образом приспособиться к окружающей среде. Схема «выживания сильнейшего» способствует выделению и усилению тех признаков, которые позитивно влияют на приспособленность конкретного индивида для решения поставленной задачи.

Особенностью ГА является то, что в поиске оптимума ведется обработка не одного вектора решений, а целого множества таких векторов, составляющих так называемую популяцию. По сути, это означает параллельный поиск решения задачи сразу по нескольким направлениям.

ГА превосходят МСП тем, что в процессе работы учитывают уже проанализированные решения и комбинируют их для получения новых, осуществляя не случайные блуждания в пространстве решений, а флуктуационно-направленное смещение в сторону оптимума. Методы МГС чаще всего проигрывают ГА потому, что дают хорошие результаты только в случае унимодальных (имеющих один максимум или минимум) задач, в то время как пространство решений большинства реальных задач имеет сложный ландшафт. По этой причине МГС имеют высокую вероятность схождения к локальному, а не к глобальному оптимуму и требуется применение нетривиальных способов учета этой возможности. ГА же за счет параллельно идущей эволюции целого ряда индивидов осваивают сразу некоторое множество регионов пространства решений и имеют большие шансы нахождения глобального решения.

В большинстве задач оптимизации ГА по скорости и эффективности работы не только не уступают классическим методам, но и превосходят их по устойчивости и скорости сходимости к глобальному максимуму (В. М. Курейчик, Н. И. Матвиенко) [5].

ГА имеют потенциал нахождения многих Парето-оптимальных решений за одну итерацию симуляции из-за присущего им встроенного параллелизма (E. Zitzler) [6]. По этой причине также можно организовать одновременную обработку данных задачи в сети, проведя ее распараллеливание по коллективу рабочих станций. Таким образом, за одно и то же затраченное время получается решение лучшего качества,

затраченное время получается решение лучшего качества, либо при неизменном качестве уменьшается общее время оптимизации (В. М. Курейчик, А. М. Никифоров, D. Brittain) [5, 7].

ГА успешно применяются для решения сложных задач во многих областях науки и техники (D. Brittain), однако ключом к эффективной работе алгоритма является, как было показано Голдбергом (D. Goldberg), такая схема кодирования фенотипа, при которой плотно связанные гены будут находиться в хромосоме близко друг к другу [7, 8]. Задача распределения данных и работ в сети является именно той задачей, в которой плотно связанные гены (например, указывающие на факт наличия зеркальных копий БД на серверах сети) могут находиться на разных концах хромосомы. Это будет способствовать тому, что строительные блоки с высоким фитнесом будут с высокой вероятностью разбиваться операторами кроссовера и мутации, а следовательно, ГА не сможет эффективно функционировать. По этой причине предлагается при кодировании карты сети в структуре хромосомы использовать такие методы как «случайные ключи» (random keys – RK) (F. Rothlauf, D. Goldberg, A. Heinzl) и «изучение плотной связности» (tight linkage learning – TLL) (G. Harik) [9, 10].

ЛИТЕРАТУРА

1. Листопад Н. И. Теоретические основы моделирования корпоративных информационно-телекоммуникационных систем. // Мн., 2001
2. Цвирун А. Д., Топцев К. С. Оптимизация распределения вычислительных работ и баз данных в сети ЭВМ // Препринт. М., ИПУ, 1983. 58 с.
3. Демиденко О. М., Максимей И. В. Проектное моделирование вычислительного процесса в локальных вычислительных сетях // Бел. наука. Мн., 2001. 251 с.
4. Сиротюк В. О. Разработка моделей, методов и инструментальных средств анализа и синтеза оптимальных структур баз данных в автоматизированных информационно-управляющих системах. // Дисс. д.т.н. М. 1999.
5. Курейчик В. Г. Генетические алгоритмы. Состояние. Проблемы. Перспективы // Теория и системы управления. 1999. №1. С.144–160.
6. Zitzler E. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications // PhD dissertation. Swiss Federal Institute of Technology Zurich. 1999.
7. Brittain D. Optimization of the Telecommunications Access Network // PhD dissertation. The University of Bristol. 1999.
8. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms / D. E. Goldberg, K. Deb, H. Kargupta, G. R. Harik // Proceedings of the Fifth International Conference on Genetic Algorithms. San Mateo, CA. 1993. P. 56–64.
9. Rothlauf F., Goldberg D. E., Heinzl A. Network Random Keys – A Tree Network Representation Scheme for Genetic and Evolutionary Algorithms // IlliGAL Report No. 200031. Illinois Genetic Algorithms Laboratory. University of Illinois. 2000.
10. Harik G. R., Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty Using Genetic Algorithms // PhD dissertation. The University of Michigan. 1997.

