

## ОПЫТ РАЗРАБОТКИ СЕМЕЙСТВА КЛАСТЕРНЫХ КОМПЛЕКСОВ ИНПАРКОМ

**Ключевые слова:** кластерный комплекс, схема организации параллельных вычислений, математическая модель задачи, хост-процессор, операционная среда.

### ВВЕДЕНИЕ

В настоящее время требования к производительности компьютеров возросли настолько, что для решения сверхсложных задач используют многоядерные процессоры [1]. Именно их задействуют для технологического прорыва в энерго- и ресурсосберегающих технологиях, в проектировании современных объектов и сооружений. Для суперкомпьютеров, насчитывающих от нескольких сот до двухсот тысяч процессоров, необходимы новые приемы программирования параллельных вычислений, способные организовать одновременно работу всех процессоров и минимизировать обмены между ними. Создание таких алгоритмов и программ в составе отработанных схем организации параллельных вычислений требует высочайшей квалификации и опыта программистов и стало актуальным направлением исследований.

Свойства классов задач сверхбольшой размерности:

— значительная длительность вычислений, чем обусловлено применение механизма контрольных точек, поскольку наработка на отказ аппаратуры комплекса остается достаточно низкой (6–7 часов);

— обязательное исследование наследственной погрешности и вычислительной погрешности, накапливаемой в ходе длительных численных расчетов, чем обусловлено использование программно эмулируемой арифметики повышенной точности;

— принципиальная невозможность применения технологии сервера баз данных в составе параллельно-конвейерных схем вычислений (у сервера баз данных сериализация, как способ распараллеливания процессов, построена на очереди запросов, поэтому в кластерном комплексе необходимы новые методы обработки данных большого объема).

Цель настоящей статьи — сфокусировать внимание на основных проблемах создания параллельных алгоритмов и программ для решения так называемых задач трансвычислительной сложности и охарактеризовать результаты одного подхода к решению проблемы, полученные при реализации семейства кластерных комплексов Инпаркком.

### ОРГАНИЗАЦИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ НА КЛАСТЕРНЫХ КОМПЛЕКСАХ

С конца 1970-х гг. стала актуальной разработка основ теории и программного обеспечения параллельных вычислений. Однако физически на одном процессоре существовавших в то время майнфреймов сложно было достичь значительного повышения производительности вычислений. Настоящего параллелизма вычислений на нескольких или многих скоординированно работающих процессорах достигли лишь с применением сетевых технологий. Произошло это в середине 1990-х гг., а промышленное значение приобрело в конце 1990-х — начале 2000-х гг.

Решение нашли, только изменив подход к архитектуре суперкомпьютеров кластерного типа, обеспечивающих параллельные вычисления на узлах сублокальной компьютерной сети, в каждом из которых используют процессоры массового производства, а узлы связывают в сеть высокопроизводительными каналами связи. В качестве общесистемного программного обеспечения практически повсеместно применяют свободное программное обеспечение с открытым кодом, доступное через Интернет. Эта архитектура, названная «Беовульф», быстро заняла ведущее место в мире, а образцы таких суперкомпьютеров занимают почти весь перечень «ТОР-500».

© И.Н. Молчанов, О.Л. Перевозчикова, А.Н. Химич, 2009

Появилась возможность построения суперкомпьютеров для всех желающих при наличии финансовых ресурсов. Вложив сначала некоторые средства в базовую структуру суперкомпьютера, можно многократно увеличивать его мощность, не заменяя, а лишь пополняя его компоненты. За 2005–2008 гг. в Институте кибернетики им. В.М. Глушкова НАНУ, кроме комплексов СКИТ, совместно с киевским ГНПО «Электронмаш» созданы опытные образцы рабочих станций Инпарком-16, -32, -64, -128, -256. По западным меркам, мощность комплекса Инпарком-256 соответствует кластеру университетского уровня. Фактически отечественные ММД-компьютеры воплотили выдвинутую В.М. Глушковым в конце 1970-х гг. концепцию параллельно-конвейерных вычислений.

Для кластерных комплексов необходимы интеллектуальные (построенные на знаниях) ИТ, требующие фундаментальных исследований логико-алгоритмического аппарата, современных операционных сред и инструментов разработки программ. По сути, для разработки, например, MPI-программ нужна принципиально новая парадигма программирования. Быстрое продвижение Института кибернетики в создании инструментария параллельных программ и получение им нетривиальных результатов обусловлены достижениями Киевской школы теории программирования, для которой направления фундаментальных исследований всегда совпадали с интеллектуализацией процесса проектирования программ.

За последнее десятилетие вследствие существенного прогресса в проектировании программного продукта, а также благодаря разнообразным методологиям быстрой разработки приложений и объектно-ориентированным CASE-системам программирования стали рассматривать как системную интеграцию на основе «готовых решений». Причиной прогресса послужили формализация и стандартизация всех этапов жизненного цикла на основе структурной декомпозиции предметной области. Преобразование задачи в сеть более простых задач, потоков данных — в сеть обобщенных сообщений и транзакций, а объектов и процессов предметной области — в совокупность интерфейсов обеспечило иерархическое распределение задач, полномочий и ответственности участников проекта с одновременным расслоением работ между ними. Консорциум OMG считает приоритетной разработку подобных средств в составе инструментария UML и других форм спецификации программного продукта.

Успешно воплотить комплексную методологию можно только при условии идеальной разработки на каждом уровне иерархии. На самом деле, большинство специалистов в предметной области могут сформулировать свои информационные потоки и потребности в целом, но не способны провести их декомпозицию согласно требованиям компьютерного воплощения. Поэтому даже при условии относительной стабильности самих объектов компьютерного моделирования рабочие модели постоянно изменяются, что фактически сводит на нет все преимущества проведенной декомпозиции. Расход времени на создание корректной модели ненамного отличается от временных затрат на реализацию действующего программного продукта.

Особый интерес разработчиков вызывают средства программирования, не требующие построения единой централизованной модели программной системы, а поддерживающие автоматическое согласование ее фрагментов, параллельно разработанных специалистами по спиральной модели жизненного цикла. В частности, следует перейти от иерархической к сетевой архитектуре программного продукта.

Необходимо сократить путь от модели до действующей программной системы, чтобы предотвратить неэффективный расход программистского труда и средств на реализацию модели, которая успевает устареть за время реализации. Поэтому актуальны средства автоматизации, способные непосредственно переработать модель на действующий прототип системы с возможностью его дальнейшего совершенствования. Кроме средств компонентного программирования, новые подходы к автоматизации программирования необходимо обязательно гармонизировать с современными технологиями и методами разработки, т.е. развивать их эволюционно, а не революционно, не требовать радикальных изменений существующих взглядов и переобучения разработчиков.

Такую парадигму компонентного (аспектного) программирования применяют для последовательно выполняемых программ, поэтому разработчики, воплощая

программную составляющую ИТ на макроконвейере кластерного комплекса с распараллеливанием вычислений, нуждаются в существенно других методах и моделях. К тому же MPI-программы пишут для классов задач сверхбольшой размерности, требующих обработки супербольших объемов данных. Вычислительную сложность алгоритмов таких ИТ можно преодолеть двумя путями:

— встраивая в математическую модель конвейеризацию обработки и связанную с ней схему распределения памяти периода вычислений, привязанную к архитектуре компонентов кластера и топологии межсоединений его узлов; тогда динамику структурных изменений в MPI-программе в течение решения классов задач можно исследовать посредством доказательства адекватности модели, прежде всего ее непротиворечивости;

— возложив на сам кластер организацию преобразования последовательной программы в параллельную и отслеживая принципиальные расхождения во время функционирования MPI-программы.

Речь идет об интеллектуализации самих инструментов проектирования программного продукта, применение которых существенно повысит интеллект создаваемых ИТ. Тем не менее в вычислительной среде кластерных комплексов имеем ряд непреодолимых проблем удобной поддержки обычной отладки программ или организации не сугубо пакетного, а интерактивного режима решения задачи (функционирование MPI-программы), когда достаточно простое взаимодействие с пользователем содействовало бы экономии времени или других ресурсов.

На основании опыта внедрения ИТ необходимо сформулировать требования к качеству программной составляющей ИТ и инструментов разработки параллельных MPI-программ в операционной среде кластерного комплекса. На пути повышения интеллектуальности средств и инструментов разработки программного продукта, способных сократить и упростить программирование параллельных программ, целесообразно задействовать:

— объемный набор re-use-компонентов повторного применения, используемый как «готовые решения»;

— удобные средства отладки параллельных MPI-программ с развитым аппаратом обработки прерываний для продолжения «жизни комплекса»;

— методы и механизмы балансирования нагрузки узлов кластера, в первую очередь многоядерных;

— математические модели и автоматизированные общесистемные средства распределения памяти (периода исполнения программ, в долгосрочных хранилищах сверхбольшого размера), способные адаптироваться к изменению архитектуры кластеров и примененных топологий сетевого соединения узлов.

#### ТЕХНОЛОГИЯ РЕШЕНИЯ ЗАДАЧ

Для постановки и решения задач науки и инженерии их предварительно исследуют в целях создания и обоснования необходимого алгоритма решения, далее пишут и отлаживают программу. Схема исследования и решения задач на компьютере приведена на рис. 1.

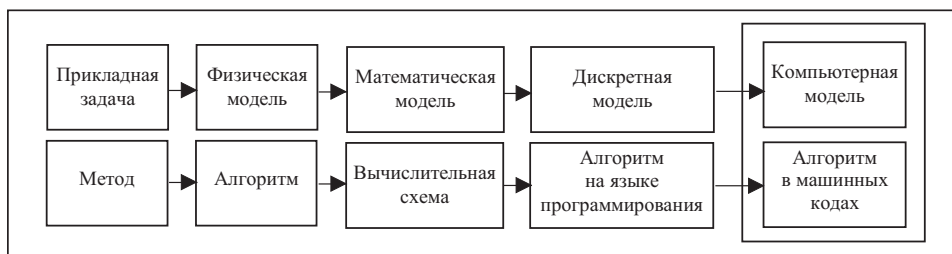


Рис. 1

В конечном итоге для решения новых задач средней сложности требуется 2–3 года, а для сложных задач — 4–5 лет.

Иногда получают компьютерные решения, не обладающие физическим смыслом. Покажем на элементарном примере одну из причин этого явления. Свободные члены систем уравнений

$$\begin{aligned} 100x_1 + 500x_2 &= 1700, \\ 15x_1 + 75,01x_2 &= 255, \end{aligned} \quad (1)$$

$$\begin{aligned} 100x_1 + 500x_2 &= 1700, \\ 15x_1 + 75,01x_2 &= 255,03 \end{aligned} \quad (2)$$

отличаются в пятой значащей цифре. Решением первой из них является  $x_1 = 17$ ,  $x_2 = 0$ , а второй —  $x_1 = 2$ ,  $x_2 = 3$ . Определитель этих систем равен единице.

Увеличив погрешность задания исходных данных, например, для

$$\begin{aligned} 100x_1 + 500x_2 &= 1700, \\ 15x_1 + 75,01x_2 &= 255,000003, \end{aligned} \quad (3)$$

получим  $x_1 = 16,9985$ ,  $x_2 = 0,0003, \dots$ , что является достаточно хорошим приближением к решению  $x_1 = 17$ ,  $x_2 = 0$  невозмущенной системы (1).

Этот пример показывает влияние наследственной погрешности на достоверность полученных результатов. Отметим, что все инженерные и научные задачи имеют приближенные исходные данные, которые не учитываются в современных программных средствах. Влиянию других факторов на точность получения решения посвящены работы [2–4]. Только зная свойства компьютерной модели задачи, можно оценить достоверность полученных решений.

Таким образом, возникает потребность использования компьютера для исследования свойств математических, дискретных и компьютерных моделей задач.

Известные теоремы классической математики невозможно применить для компьютерного исследования математических задач. Так, для выбора метода решения систем линейных алгебраических уравнений (СЛАУ) необходимо выяснить, совместна система или нет. Ответ содержится в теореме Кронекера–Капелли: «Система линейных алгебраических уравнений совместна тогда и только тогда, когда ранг матрицы равен рангу расширенной матрицы». Но для вычисления ранга матрицы системы, насчитывающей несколько тысяч или десятков тысяч неизвестных, не хватит вычислительных ресурсов, а кроме того, из-за погрешности машинной реализации ранг будет вычислен неправильно. Необходим компьютерный аналог данной теоремы, дающий ответ на вопрос о совместности компьютерных моделей СЛАУ.

В связи с этим разработана компьютерная методология исследования свойств задач вычислительной математики с приближенно заданными исходными данными, с учетом математических и технических особенностей вычислительной среды компьютеров, в том числе и MIMD-компьютеров. На основе этой методологии создано интеллектуальное численное программное обеспечение [5–11], апробированное в 1996–2002 гг. в проектах ISPAR [12] и ISKON [13] для Немецкого центра по авиакосмическим полетам DLR.

Знание свойств решаемых задач вычислительной математики, к которым сводится 90% всех инженерных и научных задач, дает возможность не только оценить достоверность получаемых решений, но и существенно сократить время решения задач [12–16]. Сокращение времени достигается за счет алгоритмов, учитывающих свойства решаемых задач, вычислительных схем, учитывающих архитектуру процессоров и топологию параллельного компьютера, а также возможности перестройки топологии из процессоров кластерных компьютеров («линейка», «решетка», «кольцо», «ветвящееся дерево» и т.д.).

Таким образом, компьютер и операционная среда должны обладать следующими свойствами:

- поддерживать интеллектуальное численное программное обеспечение;
- иметь автоматически перенастраиваемую топологию параллельного компьютера в зависимости от свойств задачи.

## ОПИСАНИЕ СЕМЕЙСТВА ИНПАРКОМ

Таким требованиям удовлетворяет созданное совместно Институтом кибернетики им. В.М. Глушкова НАНУ и ГНПП «Электронмаш» семейство интеллектуальных рабочих станций Инпаркком (табл. 1, 2), занимающее промежуточную нишу между супер- и персональным компьютером.

**Таблица 1**

Название	Характеристики семейства рабочих станций Инпаркком				
	16	32	64	128	256
Процессоры	Xeon Quad-Core	Xeon Quad-Core	Xeon Quad-Core	Xeon Quad-Core	Xeon Quad-Core
Количество узлов/процессоров/ядер	2/4/16	4/8/32	8/16/64	16/32/128	16/64/256
Пиковая производительность, GFlops	100–190	200–380	400–770	800–1500	2385
Производительность на LINPACK, GFlops	75–145	150–290	300–580	600–1130	1855
Оперативная память, GB	32	64	128	256	512
Дисковая память, TB	0,5	1	2	4	8
Емкость дискового хранилища, TB	—	—	От 1	От 2	От 3

Как знаниеориентированный инструмент Инпаркком хранит знания о свойствах компьютерной модели задачи и на основании этого автоматически строит алгоритм решения, формирует топологию из процессоров MIMD-компьютера и создает код программы параллельных вычислений, а по окончании вычислений оценивает достоверность полученных результатов.

В состав комплекса Инпаркком входит: хост-система, обрабатывающая часть, коммуникационная среда.

Хост-система осуществляет управление многопроцессорным вычислительным ресурсом, общесистемный мониторинг, общение с терминальными сетями пользователей, визуализацию результатов решения задачи и реализацию той части процесса вычислений и обработки данных, которая не распараллеливается («плохо» распараллеливается). Хост-система состоит из хост-компьютеров (Xeon Quad-Core GHz, 64 bit длина машинного слова, 8 GB оперативной памяти, 72 GB памяти на дисках каждый), внешнего оборудования и может входить в локальную или глобальную сеть.

Обрабатывающая часть, поддерживающая решение задачи с организацией параллельных вычислений, — это однородная масштабируемая структура, состоящая из множества процессоров (с собственной оперативной и дисковой памятью), объединенных коммуникационной средой межпроцессорного взаимодействия. Обрабатывающая часть может включать вычислительные узлы (Xeon Quad-Core GHz, 64 bit длина машинного слова, 2 GB оперативной памяти, 36 GB памяти на дисках каждый).

Коммуникационная среда состоит из Gigabit Ethernet; InfiniBand и гиперкуба. Программное обеспечение предусматривает три уровня:

— операционная среда, поддерживающая интеллектуальное программное обеспечение;

— интеллектуальное численное программное обеспечение для исследования и решения задач вычислительной математики с приближенно заданными данными;

— прикладное программное обеспечение, например для исследования и решения задач анализа прочности конструкций.

В основу операционной среды положены бесплатные решения GNU/Linux. Однако пользователь может выбрать один из трех вариантов установленных ОС: Linux, Windows XP SP2 или Linux+Windows. По желанию пользователя хост автоматически переключается на Linux и Windows с перезагрузкой узлов. Версия Linux на основе Scientific Linux 4.2 (от 22.11.2005) оптимизирована под аппаратуру комплекса Инпаркком.

В ядре параллельного компьютера — системе передачи сообщений — реализован стандарт де-факто MPI. В Linux установлен MVAPICH, оптимизированный под InfiniBand, и LAM MPI, в Windows — MPICH. Для поддержки максимального числа приложений сторонних пользователей настроена распространенная система передачи сообщений PVM.

**Таблица 2**

Название	Вычислительный узел	Графическая станция	Дисковое хранилище
	Вычисления	Управление, ввод/вывод графической информации	Управление, хранение массивов данных
<b>Технические средства</b>			
Процессор	2x2xIntel Xeon 53XX	2xIntel Xeon 51XX (53XX)	Intel Xeon 51XX
Количество ядер	8	8	2
Объем оперативной памяти	2x16 GB DDR2-667	16(32) GB DDR2-667	2 GB DDR2-667
Дисковая память	HDD 2x2x250 GB (2xRAID1)	HDD 2x250 GB (RAID1), FDD, DVD±RW	HDD 10x250 GB (2xRAID1, 8xRAID 0,5,10,50), FDD, DVD-ROM
Количество узлов	16 вычислительных узлов; две графические станции (количество определяет заказчик); два дисковых хранилища		
Вычислительная сеть	InfiniBand (20 Gbit/s)		
Служебная сеть	Gigabit Ethernet, Fast Ethernet (IPMI с KVM)		
Система бесперебойного питания	ИБП 10000VA на вычислительный блок — 2 шт., ИБП 1000VA на каждую рабочую станцию, On-line		
Конструкция	Вычислительный блок — шкаф 19" / 25U — 2 шт. Графическая станция — системный блок, монитор, клавиатура, мышь, ИБП, принтер, сканер		
<b>Системное программное обеспечение</b>			
Операционная система	Linux на базе Red Hat EL 5, Linux или Windows на графической станции		
Параллельная среда	MPI (OpenFabrics Enterprise Distribution)		
Система управления	Программный системный монитор (управление заданиями, мониторинг заданий и аппаратуры комплекса)		
<b>Интеллектуальное программное обеспечение</b>			
Библиотеки	Библиотеки интеллектуальных программ решения задач вычислительной математики с оценкой достоверности (Inparlib): — системы линейных алгебраических уравнений; — алгебраическая проблема собственных значений; — системы нелинейных алгебраических и трансцендентных уравнений; — системы обыкновенных дифференциальных уравнений с начальными условиями		
Интерфейс	Диалоговая планирующая и управляющая система решения задач вычислительной математики (Inpartool)		
Прикладное ПО	Интеллектуальное прикладное программное обеспечение для исследования и решения задач анализа прочности конструкций (на базе ПО Лира [19])		

Бесплатный компилятор GCC в составе Linux поддерживает Си/C++, Фортран и Java. Операционная среда включает Интернет-сервер Apache с поддержкой приложений на языке PHP, СУБД MySQL, стандартные математические библиотеки (в том числе ScaLAPACK), тесты (Linpack, Scali), сетевую файловую систему.

Операционная среда обеспечивает:

- формирование задания и его запуск на выбранных вычислительных узлах;
- мониторинг всего компьютера и выполняемых заданий;
- сохранение и визуализацию протоколов параллельных расчетов;
- запуск приложения (параллельной программы) на хост-компьютере;
- работу через локальную сеть и/или Интернет с удаленным доступом;
- разработку параллельных программ;
- администрирование доступных пользователю частей сетевой файловой системы.

Интеллектуальное численное программное обеспечение для исследования и решения задач вычислительной математики с приближенно заданными исходными данными под-держивает: автоматический режим полного исследования и решения задач; решение задач выбранной программой из библиотеки. Реализованы классы задач:

- системы линейных алгебраических уравнений;
- алгебраическая проблема собственных значений;
- системы нелинейных алгебраических и трансцендентных уравнений;
- системы обыкновенных дифференциальных уравнений с начальными условиями.

Прикладное программное обеспечение, например, для исследования и реше-ния задач анализа прочности конструкций может содержать средства:

- формирования геометрической модели конструкции на основе имеющихся в банке геометрических моделей;
- формирования математической модели задачи на основе моделей, имею-щихся в банке данных;
- формирования конечно-элементной или конечно-разностной модели задачи;
- визуализации конечно-элементного покрытия элемента или исследуемой конструкции;
- формирования в автоматическом режиме дискретной модели задачи и рас-сылки данных по процессорам выбранной топологии;
- обращения к интеллектуальному численному программному обеспечению для исследования и решения сформулированных конечно-элементных задач на MIMD-компьютере с визуализацией полученных результатов;
- анализа достоверности полученного конечно-элементного или конечно-разностного решения.

Таким образом, вместо традиционной схемы исследования и решения задачи (рис. 1) возникает новая схема (рис. 2), в которой компьютер вместо пользователя вы-полняет все необходимые исследования и создает топологию и программу параллель-ных вычислений.

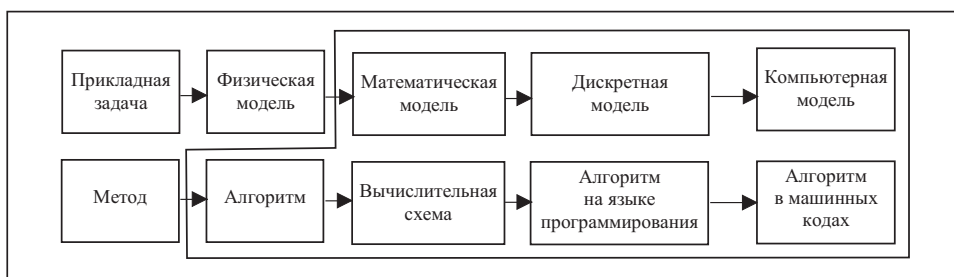


Рис. 2

Области использования комплекса Инпарком:

- численное моделирование сложных процессов, явлений, объектов и систем для организации на этой основе натуральных экспериментов;
- создание тренажеров управления сложными объектами современной техни-ки, в том числе АЭС;
- решение сложных научно-технических задач с приближенно заданными исходными данными;
- подготовка параллельных программ для суперкомпьютеров.

Способы использования:

- адаптация созданных ранее программных средств на кластерах Инпарком: реализация препроцессинга существующих программных средств для перевода зада-чи на языке пользователя в математическую задачу на хост-системе; процессинг как решение математической задачи в автоматическом режиме с параллельной организа-цией вычислений на вычислительном блоке Инпаркома, постпроцессинг — перевод решения математической задачи на язык пользователя на хост-системе;
- создание программных комплексов с учетом возможности интеллектуаль-ных рабочих станций Инпарком.

Преимущества комплекса Инпарком:

- постановка задачи пользователя на языке предметной области;

— освобождение пользователя от исследования свойств задачи и создания алгоритма решения, написания и отладки параллельной программы, что сокращает время постановки и решения задачи не менее чем в 100 раз;

— исследование и решение задач инженерии и науки с приближенно заданными исходными данными;

— получение машинного решения с оценкой достоверности, а также всех свойств решаемой машинной модели задачи с приближенно заданными исходными данными;

— существенное сокращение времени машинного исследования и решения научно-технических задач по сравнению с традиционной технологией решения той же задачи на ММД-компьютере с тем же количеством процессоров и на той же элементной базе, но с традиционной параллельной архитектурой.

#### ПРИМЕРЫ РЕШЕННЫХ ЗАДАЧ

Использование комплекса Инпарком целесообразно для математического моделирования, т.е. компьютерного решения уравнений, описывающих исследуемый объект. Оно помогает рассмотреть несколько вариантов исследуемого объекта и в результате спланировать натуральный эксперимент. Так, на Инпарком-64 промоделировано обтекание планера для АН-148, отработано натурное моделирование этого планера в аэродинамической трубе ЦАГИ. Корреляция данных численного и натурального экспериментов не превысила 3%, а сам натуральный эксперимент обошелся КБ Антонова в 3–4 раза дешевле, чем без предварительного моделирования.

На комплексе Инпарком выполнен расчет прочности конструкции двухкорпусного 80-этажного офисного центра в Москве [17] (рис. 3). Параметры расчета приведены в табл. 3, а структура матрицы представлена на рис. 4.

Для Института электросварки имени Е.О. Патона решена задача нахождения кинетики температурных напряжений и деформаций (напряженно-пластическая задача) при сварке титановых балок пола в конструкциях самолетов [18] (рис. 5). Собственно рассчитанная траектория сварки показана темной линией внизу конструкции.

**Таблица 3**

Конечно-элементная сетка		
Количество элементов	972808	
Количество узлов	895302	
Характеристики дискретной задачи		
Порядок СЛАУ	5371727	
Плотность матрицы жесткости	1%	
Полуширина ленты матрицы	3174975	
Размер файла исходных данных	76 GB	
Характеристики расчета		
Компьютер	Количество ядер	Время решения
Инпарком-256	256	4 ч 55 мин
Инпарком-256	192	5 ч 29 мин
Инпарком-128	128	6 ч 09 мин
Инпарком-64	64	9 ч 24 мин
ПК ЛИРА 9.2	1	Примерно 115 ч

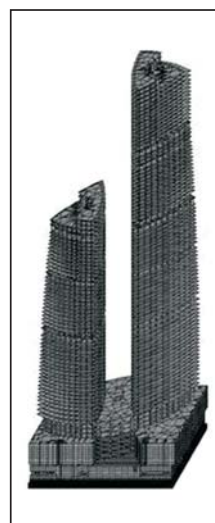


Рис. 3

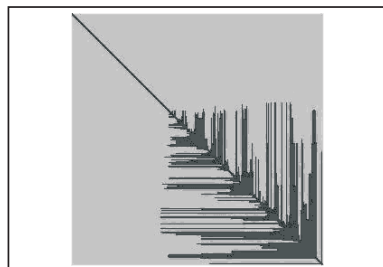


Рис. 4

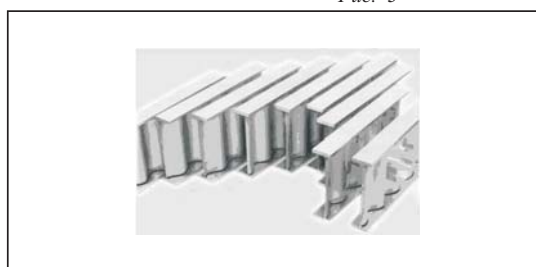


Рис. 5



## ЗАКЛЮЧЕНИЕ

На начало 2009 г. в Украине работали 17 кластерных комплексов, среди которых вместе с семействами СКИТ и Инпарком следует назвать кластеры НТУУ «Киевский политехнический институт», Института теоретической физики НАН Украины, Главной астрономической обсерватории НАН Украины, Киевского национального университета имени Тараса Шевченко, Института сцинтиляционных материалов НАН Украины (Харьков). Можно констатировать, что своевременно начатое направление кластерных вычислений приобрело в Украине мощное развитие.

## СПИСОК ЛИТЕРАТУРЫ

1. <http://www.top500.org>.
2. Молчанов И. Н. Машинные методы решения задач прикладной математики. Алгебра, приближенные функции, обыкновенные дифференциальные уравнения. — Киев: Наук. думка, 2007. — 550 с.
3. Уилкинсон Дж. Алгебраическая проблема собственных значений. — М.: Наука, 1970. — 564 с.
4. Воеводин В. В. Ошибки округления и устойчивость прямых методов линейной алгебры. — М.: Изд-во МГУ, 1969. — 143 с.
5. Молчанов И. Н. Информационно-вычислительная технология решения задач // Кибернетика и системный анализ. — 1994. — № 6. — С. 65–72.
6. Молчанов И. Н., Химич А. Н., Чистякова Т. В. Алгоритмические и вычислительные возможности интеллектуального программного средства LINSYST // Там же. — 1998. — № 3. — С. 40–50.
7. Молчанов І. М., Гець О. Г., Герасимова Т. О., Яковлев М. Ф. Інтелектуальна система дослідження та чисельного інтегрування задач Коші для систем звичайних диференціальних рівнянь // Теорія обчислень: Збірка наук. праць НАНУ. — Київ: Ін-т кібернетики НАНУ, 1999. — С. 260–264.
8. Інтелектуальний інтерфейс для дослідження та розв'язування задач обчислювальної математики з наближено заданими вхідними даними на MIMD-комп'ютері / І.М. Молчанов, Є.Ф. Галба, О.В. Попов, О.М. Хіміч, Т.В. Чистякова, М.Ф. Яковлев // Проблеми програмування. — 2000. — № 1/2. — С. 102–112.
9. Молчанов И. Н., Чистякова Т. В. Интеллектуальное программное обеспечение для исследования и решения прикладных задач с приближенно заданными исходными данными // УСиМ. — 2003. — № 3. — С. 72–76.
10. Интеллектуальная система для исследования и решения матричных задач на собственные значения / И.Н. Молчанов, Т.А. Герасимова, О.В. Попов и др. // Проблемы программирования. — 2004. — № 2–3. — С. 570–576.
11. Молчанов И. Н. Машинная математика — проблемы и перспективы // Кибернетика и системный анализ. — 2004. — № 6. — С. 65–72.
12. Intelligente Umgebung zur Untersuchung und Loesung wissenschaftlich-technischer Aufgaben auf Parallelrechnern, 1996–1998 (Федерал. Министерство образ., науки, исследований и технологии Германии; ISPAR, 01 IR 601/8).
13. Intelligente Software zur Untersuchung und Loesung von Aufgaben zur Analyse der Festigkeit von Konstruktionen, 1999–2002 (Федерал. Министерство образ., науки, исследований и технологии Германии; ISKON, 01 IR 905/3).
14. Moltschanow I. N., Heuer H., Stuchlik F. Problems encountered with the development of intelligent MIMD-computers. — Magdeburg, 1995. — (Prepr. / Fak. für Informatik Otto von-Guericke Univ.; N 8).
15. Молчанов И. Н. Интеллектуальные компьютеры — средство исследования и решения научно-технических задач // Кибернетика и системный анализ. — 2004. — № 1. — С. 175–179.
16. Численное программное обеспечение интеллектуального MIMD-компьютера Инпарком / А.Н. Химич, И.Н. Молчанов, В.И. Мова, О.Л. Первозчикова и др. — Киев: Наук. думка, 2007. — 222 с.
17. Химич А. Н., Полянок В. В., Попов А. В., Рудич О. В. Решение задач расчета прочности конструкций на MIMD-компьютере // Искусств. интеллект. — 2006. — № 4. — С. 138–147.
18. Математическое моделирование на MIMD-компьютерах физических процессов при сварке / В.И. Махненко, А.В. Попов, А.П. Семенов, А.Н. Химич, М.Ф. Яковлев // УСиМ. — 2007. — № 6. — С. 80–87.
19. ЛИРА 9.2. Руководство пользователя. Основы: Учеб. пособие / Е.Б. Стрелец-Стрелецкий, Ю.В. Гензерский, Д.В. Марченко и др.; Под ред. А.С. Городецкого. — Киев: ФАКТ, 2005. — 146 с.

*Поступила 08.07.2009*