

Рис. 5. Моделирование эквивалентной деформации элементов нанопинцета

В настоящее время разработки в области нанотехнологий ведутся по всему миру практически во всех отраслях промышленности. Это подтверждает актуальность проблемы создания наноманипуляторов в сочетании с визуализацией в реальном времени, имеющих более высокое разрешение по сравнению со стандартными манипуляторами.

Литература

1. N. Taniguchi, "On the Basic Concept of 'Nano-Technology'", Proc. Intl. Conf. Prod. Eng. Tokyo, Part II, 1974, Japan Society of Precision Engineering.
2. C.D Howe. Nanotechnology: Slow Revolution. Forrester Research Corporation, August 2002, Cambridge, Maryland, USA.
3. 王作斌刘兰娇董莉彤翁占坤宋正勋范思哲, Nano-manipulation system and method for three-point probe robot // Патент КНР.

УДК 681.32.5

ПРИЛОЖЕНИЕ ПО СОЗДАНИЮ РАСПРЕДЕЛЕННЫХ СИСТЕМ НА БАЗЕ ОРИЕНТИРОВАННОГО ПОДХОДА

студент гр. 10306119 Шункевич Е.А.

Научный руководитель - к.т.н., доцент Гулай А. В.

Белорусский национальный технический университет
Минск, Беларусь

Введение

В данной научной работе рассматриваются проблемы проектирования и создания распределенных информационно-управляющих систем (ИУС), которые осуществляют сбор, обработку и хранение потоковых данных, взаимодействуют с реальной средой посредством территориально распределенных датчиков и исполнительных механизмов.

Современные программно-аппаратные средства автоматизации не позволяют организовать гибкое распределение вычислительных ресурсов между подзадачами и при этом обеспечивать живучесть и надежность создаваемого технического обеспечения ИУС. Предлагается агентно-ориентированная программная архитектура систем обработки потоковых данных, которая позволяет обеспечить гибкость и живучесть программного обеспечения распределенных ИУС.

Эффективная организация взаимодействия между разработчиками различных подсистем обеспечивает снижение ресурсозатрат на создание и эксплуатацию системы. Наиболее известным программным решением для распределенных программных систем является использование трехзвенных программных архитектур «клиент-сервер» и создание дополнительного слоя программного обеспечения - промежуточного слоя, который позволяет сократить затраты на проектирование и разработку программных модулей обработки, хранения данных, а также их интеграцию. Применительно к распределенным системам управления применимость такого подхода осложняется спецификой требований, предъявляемых к техническому обеспечению ИУС: требования по безопасности, надежности и помехозащищенности оборудования, сетей связи, а также протоколов передачи данных между узлами ввода-вывода и обработки потоковых данных.

Постановка задачи

Задачу обработки потоковых данных можно представить в виде совокупности множества атрибутов и множества процедур их преобразования. Пусть задано множество атрибутов V , значения которых необходимо поддерживать актуальными. Подзадачи обработки данных в этом случае можно определить множеством операций F по вычислению актуальных значений одной совокупности атрибутов с использованием значений другой совокупности атрибутов.

Каждая такая операция может быть представлена кортежем:

$$f \in F; \quad f = \langle V_G^f, V_D^f \rangle$$

где $V_G^f, V_D^f \subseteq V$; $V_G^f \cap V_D^f = \emptyset$ – множество входных и выходных атрибутов для данной операции.

Поток данных между двумя операциями можно представить следующим кортежем: $c \in C$; $c = \langle f_S^c, V_R^c \rangle$.

Модель обработки потоковых данных может быть представлена в виде ориентированного биграфа (рис. 1):

$$G = \langle N, A \rangle$$

где: $N = V \cup F$ – множество вершин графа; им соответствует объединение множеств атрибутов и процедур; $A \subseteq V * F$ множество упорядоченных пар вершин.

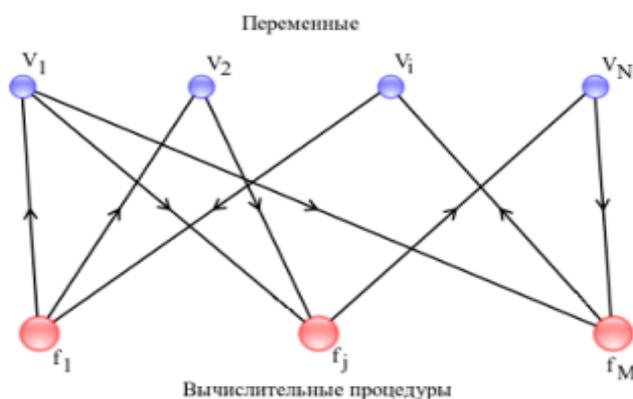


Рис. 1. Представление задачи обработки данных в виде двудольного орграфа операций и атрибутов

Решение

Вершины типа f , в которые не входит ни одной дуги, соответствуют операциям, которые можно представить источниками потоковых данных. Они соответствуют процедурам взаимодействия с оператором опроса или оператором опроса датчиков. Аналогично, операцию отображения или сохранения потоковых данных можно считать приемником данных. Им соответствуют те вершины f графа, из которых не исходит ни одной дуги.

Подобное представление обуславливает специфическую структуру программных компонент; реализующих задачу обработки потоковых данных, и позволяет поставить задачу создания агентно-ориентированной программной архитектуры. Необходимо разработать структуру программных компонентов и модель взаимодействия между ними, которые бы позволяли создавать агентные программные системы распределенной обработки потоковых данных, обеспечивать их гибкость, масштабируемость и живучесть, а также ставить и решать задачу распределения ресурсов гетерогенных вычислительных сетей.

При исследовании функциональной живучести программных систем особенности топологии межкомпонентных связей учитываются опосредовано. Предполагается, что в программных системах обеспечивается необходимая связность работоспособных компонентов. С точки зрения живучести программная система характеризуется следующими параметрами:

целью функционирования;

множеством задач $Q = \{q_1, q_1, \dots, q_m\}$, решение которых обеспечивает ее достижение;

множеством компонент $S = \{s_1, s_1, \dots, s_p\}$, являющихся составными частями системы.

Принципиальное отличие понятия отказоустойчивости от живучести, применительно к программным системам, показанных на рис.2. На графиках показана зависимость условного количества решаемых задач (ось ординат) от условного времени жизненного цикла системы (ось абсцисс).

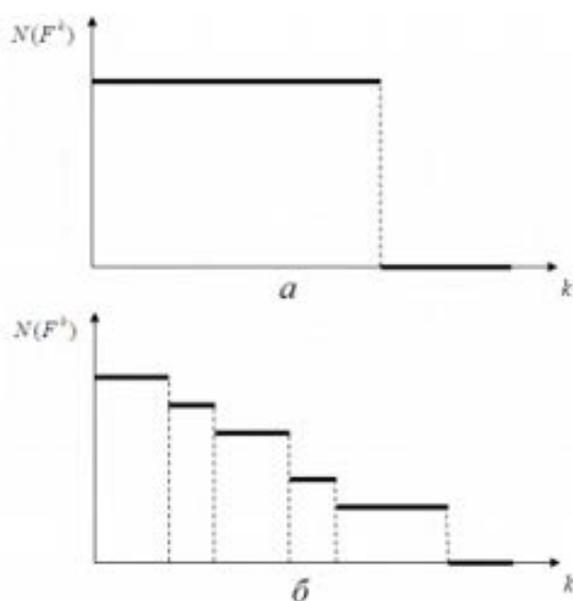


Рис. 2. Зависимость количества решаемых задач от времени функционирования системы в условиях обеспечения: а) отказоустойчивости; б) живучести системы