**Введение в методы параллельного программирования**

* MPI: основные понятия и определения
* Введение в MPI
  + Инициализация и завершение MPI программ
  + Определение количества и ранга процессов
  + Прием и передача сообщений
  + Определение времени выполнения MPI программы
  + Коллективные операции передачи данных
* Пример: программа вычисления числа PI
* Заключение

В вычислительных системах с распределенной памятью процессоры работают независимо друг от друга.

Для организации параллельных вычислений необходимо уметь:

* Распределять вычислительную нагрузку
* Организовать информационное взаимодействие между процессорами
* В рамках MPI для решения задачи разрабатывается одна программа, она запускается на выполнение одновременно на всех имеющихся процессорах
* Для организации различных вычислений на разных процессорах:
  + Есть возможность подставлять разные данные для программы на разных процессорах
  + Имеются средства для идентификации процессора, на котором выполняется программа
* Такой способ организации параллельных вычислений обычно именуется как модель “одна программа множество процессов”
* В MPI существует множество операций передачи данных:
  + Обеспечиваются разные способы пересылки данных
  + Реализованы практически все основные коммуникационные операции

Эти возможности являются наиболее сильной стороной MPI

Что означает MPI?

* MPI – это стандарт, которому должны удовлетворять средства организации передачи сообщений
* MPI – это программные средства, которые обеспечивают возможность передачи сообщений и при этом соответствуют всем требованиям стандарта MPI:
  + Программные средства должны быть организованы в виде библиотек программных модулей
  + Должны быть доступны для наиболее широко используемых алгоритмических языков C и Fortram

Достоинства

* MPI уменьшает сложность разработки параллельных программ:
  + Большая часть основных операций передачи данных предусматривается стандартом MPI
  + Имеется большое количество библиотек параллельных методов, созданных с использованием MPI
* MPI содействует повышению эффективности параллельных вычислений – практически для каждого типа вычислительных систем существуют реализации библиотек MPI

История разработки MPI

* 1992 г. Начало работ над стандартом библиотеки передачи сообщений
* Ноябрь 1993 г. Объявление рабочего варианта стандарта MPI 1.
* Ноябрь 1993 г. Обсуждение стандарта на конференции Supercomputing’93.
* 5 мая 1994 г. Окончательный вариант стандарта MPI 1.0.
* 12 Июня 1995 г. Новая версия стандарт – MPI 1.1.
* 18 Июля 1997 г. Опубликован стандартом MPI-2.

Понятие параллельной программы

* Под параллельный программной в рамках MPI понимается множество одновременно выполняемых процессов:
* Процессы могут выполняться на разных процессорах; вместе с этим на одном процессоре могут располагаться несколько процессов
* Каждый процесс параллельной программы порождается на основе копии одного и того же программного
* В основу MPI положены четыре основные концепции:
* Тип операции передачи сообщения
* Тип данных, пересылаемых в сообщении
* Понятие коммуникатора
* Понятие виртуальной топологии

Операции передачи данных

* Основу MPI составляют операции передачи сообщений.
* Среди предусмотренных в составе MPI функций различаются:
  + Парные операции между двумя процессами
  + Коллективные коммуникационные действия для одновременного взаимодействия нескольких процессов

Понятие коммутаторов

* В ходе вычислений могут создаваться новые и удаляться существующие коммуникаторы
* Один и тот же процесс может принадлежать разным коммуникаторам
* Все имеющиеся в параллельной программе процессы входят в состав создаваемого по умолчанию коммуникатора с идентификатором MPI\_COM\_WORLD
* При необходимости передачи данных между процессами из разных групп необходимо создавать глобальный коммуникатор

**Методология информационного моделирования IDEF1X**

Основные вопросы:

* Основные понятия: сущность, атрибут, отношение
* Правила определения сущности, атрибута, отношения
* Основные правила формирования информационной модели
* Пример IDEF1X-модели на примере процесса постройки садового домика
* *Методология IDEF1X* (IDEF1 Extended) – язык для семантического моделирования данных, основанных на концепции «*сущность-связь*». Является расширением стандарта IDEF1.
* Диаграмма «*сущность-связь*» *ERD* (*Entity-Relationship Diagram*) предназначена для разработки модели данных и обеспечивает стандартный способ определения данных и отношений между ними.
* Теоретической базой построения информационной модели является теория баз данных типа «сущность-связь».
* Согласно стандарту, основными составляющими модели IDEF1X являются:

1) люди, предметы, явления, о которых хранится информация (далее – сущности)

2) связи между этими элементами (далее – отношения)

3) характеристики этих элементов (далее – атрибуты)

* Сущность – это множество реальных или абстрактных объектов (людей, мест, событий), обладающих общими атрибутами или характеристиками.
* *Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована.*
* *Пример*. Сущность – Студент. Экземпляр сущности – студент Иванов И.И.
* Атрибут – характеристика сущности.
* Пример. Сущность «Студент» имеет атрибут «ФИО».
* Экземпляр сущности «студент» (конкретный человек) будет иметь экземпляр атрибута «ФИО» (например, Иванов И.И.)
* Отношения – связь между двумя и более сущностями. Именование отношения осуществляется с помощью грамматического оборота глагола (имеет, определяет, …).

Таким образом…

* Сущности представляют собой базовый тип информации, хранимый в БД, а отношения показывают, как эти типы данных взаимосвязаны друг с другом.

1. Сущность должна иметь уникальное имя и именоваться существительным в единственном числе.

*Пример*: *Студент, Кредитная карта, Договор, …*

1. Сущность обладает одним или несколькими атрибутами, которые ей либо принадлежат, либо наследуются через отношения.
2. Сущность обладает одним или несколькими атрибутами, которые однозначно идентифицируют каждый образец сущности и называются ключом (составным ключом).
3. Каждая сущность может обладать любым количеством отношений с другими сущностями.
4. Если внешний ключ целиком используется в составе первичного ключа, то сущность является зависимой от идентификатора.

В нотации IDEF1X сущность изображается в виде прямоугольника, в зависимости от уровня представления данных могут быть некоторые различия

Различают следующие уровни представления сущности: диаграмма «сущность-связь» (*ERD*), модель данных, основанная на ключах (*KB*), полная атрибутивная модель (*FA*)

1. Каждый атрибут каждой сущности обладает уникальным именем.
2. Сущность может обладать любым количеством атрибутов.
3. Различают собственные и наследуемые атрибуты. Собственные атрибуты являются уникальными в рамках модели. Наследуемые передаются от сущности-родителя при определении идентифицирующей связи.
4. 1) При определении отношения типа *«родитель-потомок»*:
5. 1.1. Экземпляр потомка связан с одним родителем
6. 1.2. Экземпляр-родитель может быть связан с несколькими экземплярами потомков.
7. 2) В идентифицирующем отношении сущность-потомок всегда является зависимой от идентифицирующей сущности.
8. 3) Сущность может быть связана с любым количеством других сущностей как в качестве родителя, так и в качестве потомка.
9. 4) Отношение определяется мощностью. *Мощность (кратность)* связи служит для обозначения отношения количества экземпляров родительской сущности к числу экземпляров дочерней.
10. Отношения категоризации – отношения между двумя и более сущностями, в которых каждый экземпляр одной сущности, называемой общей, связан в точности с одним экземпляром сущности, называемой сущностью-категорией.
11. Категория выделяется из общей сущности по определенному признаку.

Различают полную и неполную категоризацию

1. Сущность типа «категория» может иметь только одну общую сущность.

2. Сущность-категория, принадлежащая одному отношению категоризации, может быть общей сущностью в другом отношении категоризации

3. Сущность может являться общей в *любом количестве* отношений категоризации.

4. Атрибуты первичного ключа сущности-категории должны совпадать с атрибутами первичного ключа общей сущности.

5. Все экземпляры сущности-категории имеют одно и то же значение дискриминатора, следовательно, все экземпляры других категорий должны иметь другое значение дискриминатора.

1. Все стрелки (вход, выход, управление, механизм) функциональной модели становятся потенциальными сущностями, а функции, связывающие их, трансформируются в отношения между этими сущностями. Для этого составляется список потенциальных сущностей.

2. Число сущностей и связей в IDEF1X-модели считается трудно обозримым, если их количество превышает 25-30. Поэтому далее рассматривается совокупность сущностей и отношений для каждой функции.

3. Информационная модель функции должна позволять воспроизвести структуру документа и часть информации в нем, а также воспроизвести информацию порождаемого документа.

4. Текстовые пояснения заносятся в глоссарий или оформляются гипертекстом.

5. На основании определения типов отношений, анализа функций и дальнейшего изучения предметной области определяются атрибуты.

* Информационная модель IDEF1X
* Сущность (зависимая, независимая, общая, категории, ассоциативная, именующая, характеристическая)
* Атрибут (первичный, составной, альтернативный, потенциальный, внешний ключ, не ключевой)
* Отношение (идентифицирующее, не идентифицирующее, неспецифическое, категоризации)