Распределенные системы

Крюков Виктор Алексевич
д.ф.-м.н., профессор, советник
Института прикладной математики им М.В.Келдыша РАН
профессор кафедры системного программирования
факультета вычислительной математики и кибернетики
Московского университета им. М.В. Ломоносова

Бахтин Владимир Александрович
к.ф.-м.н., ведущий научный сотрудник
Института прикладной математики им М.В.Келдыша РАН
доцент кафедры системного программирования
факультета вычислительной математики и кибернетики
Московского университета им. М.В. Ломоносова

Введение в предмет

В курсе рассматриваются проблемы создания **распределенных систем** – систем, в которых совокупность независимых компьютеров представляется их пользователям единой объединенной системой.

Распределенная компьютерная система – совокупность связанных сетью независимых компьютеров, которая представляется пользователю единым компьютером.

Распределенная программная система – совокупность компонентов, взаимодействующих посредством обмена сообщениями.

Основной задачей распределенных систем является облегчение пользователям доступа к удаленным ресурсам и обеспечение их совместного использования.

Обсуждаются способы организации взаимодействия процессов и их доступа к оперативной памяти и файловой системе.

Излагаются принципы обеспечения надежности функционирования распределенных систем.

Примеры распределенных систем

- сеть рабочих станций,
- кластер ЭВМ,
- система обеспечения банковских операций,
- система резервирования авиабилетов,
- Интернет,
- электронная почта,
- электронная коммерция,
- социальные сети,
- интерактивные игры с множеством игроков, и т.п.

Примеры распределенных систем

Черты распределенных систем

- конкурентность
- отсутствие глобальных часов
- независимые отказы

Тенденции, определяющие развитие PC сегодня

- широкое распространение сетевых технологий
- повсеместное использование компьютинга в сочетании с желанием поддерживать мобильность пользователей в распределенных системах
- растущий спрос на мультимедийные услуги
- представление распределенных систем как утилиты

Акцент на совместном использовании ресурсов

- оборудование (принтеры, диски,...)
- данные (файлы, БД)
- сервисы (поиск в интернете, онлайн редактирование, интерактивные игры)

Ложные предположения, которые делают начинающие разработчики PC

- Сеть надежная
- Сеть безопасна
- Сеть однородная
- Топология не меняется
- Задержка нулевая
- Пропускная способность бесконечна
- Транспортные расходы нулевые
- Есть один администратор

Проблемы

- гетерогенность
- открытость
- секретность
- масштабируемость
- надежность
- конкурентность
- прозрачность
- качество обслуживания (надежность, секретность, производительность, реактивность, адаптивность)

Почему создаются распределенные системы? В чем их преимущества перед централизованными ЭВМ?

- Можно достичь такой высокой производительности путем объединения микропроцессоров, которая недостижима в централизованном компьютере.
- Естественная распределенность (банк, поддержка совместной работы группы пользователей).
- Надежность (выход из строя нескольких узлов незначительно снизит производительность).
- Наращиваемость производительности.

- Главная причина - наличие огромного количества компьютеров и необходимость совместной работы без ощущения неудобства от географического и физического распределения людей, данных и машин.

Почему нужно объединять PC в сети?

- Необходимость разделять данные.
- Преимущество разделения дорогих периферийных устройств, уникальных информационных и программных ресурсов.
- Достижение развитых коммуникаций между людьми. Электронная почта во многих случаях удобнее писем, телефонов и факсов.
- Гибкость использования различных ЭВМ, распределение нагрузки.
- Упрощение постепенной модернизации посредством замены компьютеров.

Принципы построения распределенных систем

- Прозрачность
- Гибкость
- Надежность
- Эффективность
- Масштабируемость

Прозрачность

Прозрачность расположения	Пользователь не должен знать, где расположены ресурсы
Прозрачность миграции	Ресурсы могут перемещаться без изменения их имен
Прозрачность размножения	Пользователь не должен знать, сколько копий существует
Прозрачность конкуренции	Множество пользователей разделяет ресурсы автоматически
Прозрачность параллелизма	Работа может выполняться параллельно без участия пользователя

Надежность

- Доступность, устойчивость к ошибкам (fault tolerance)
- Секретность

Производительность

- Грануллированность. Мелкозернистый и крупнозернистый параллелизм (fine-grained parallelism, coarse-grained parallelism)
- Устойчивость к ошибкам требует дополнительных накладных расходов

Масштабируемость

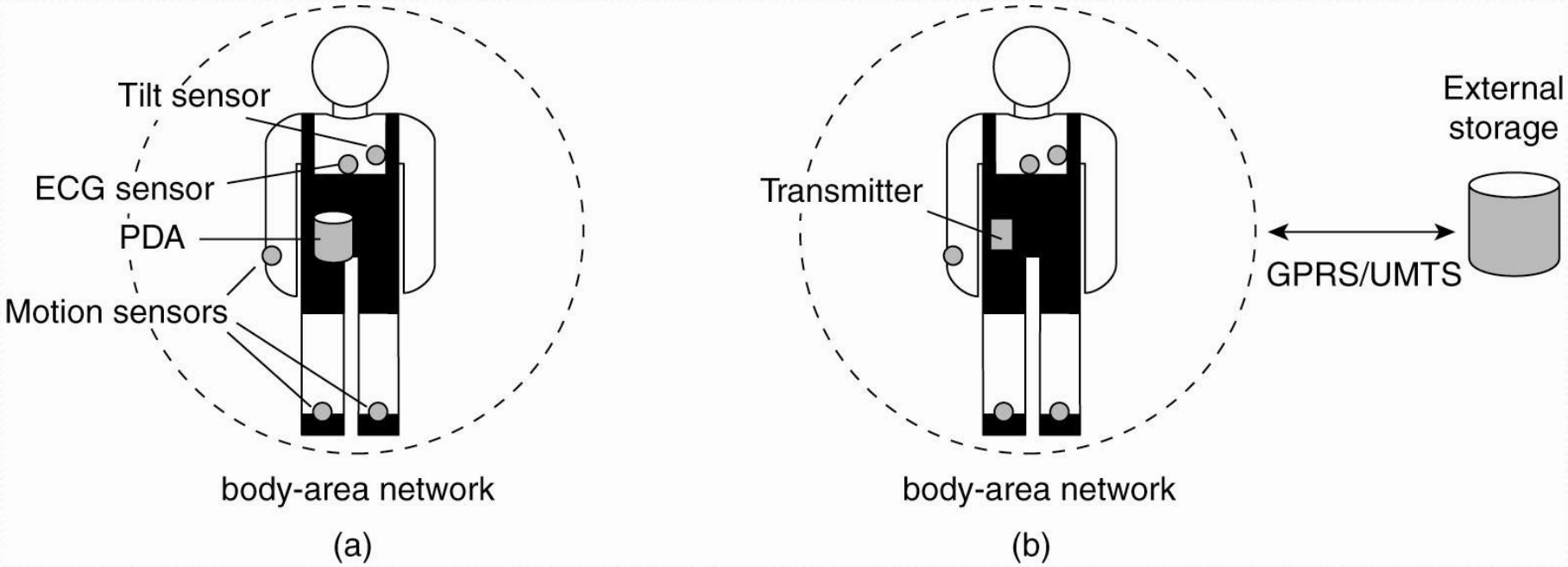
Плохие решения:

- ❑ централизованные компоненты (один почтовый-сервер);
- ❑ централизованные таблицы (один телефонный справочник);
- ❑ централизованные алгоритмы (маршрутизатор на основе полной информации).

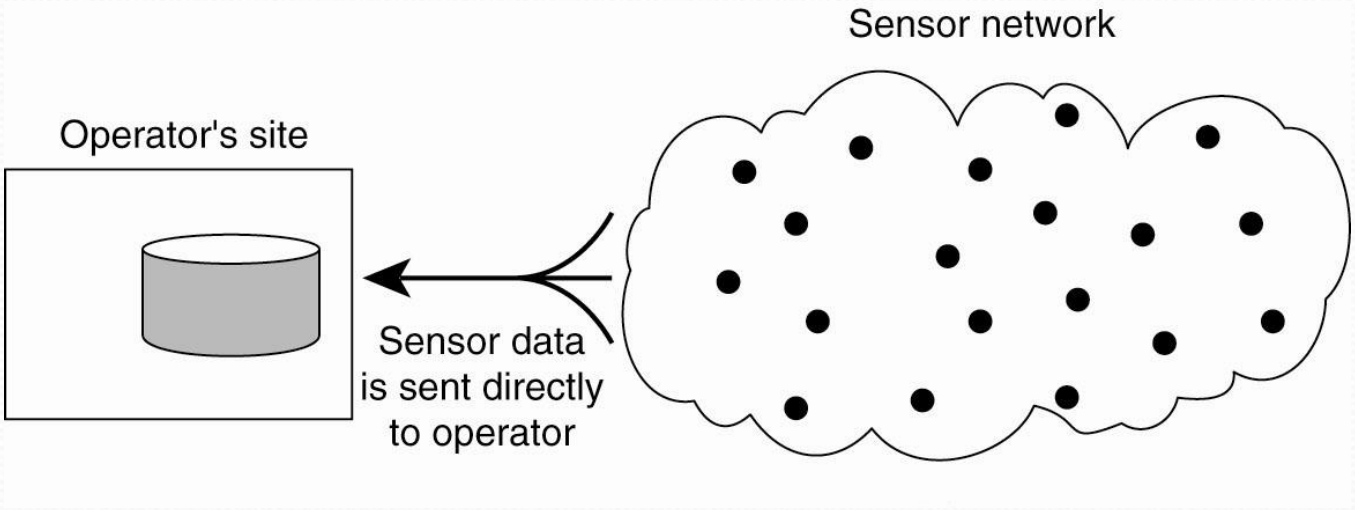
Только децентрализованные алгоритмы со следующими чертами:

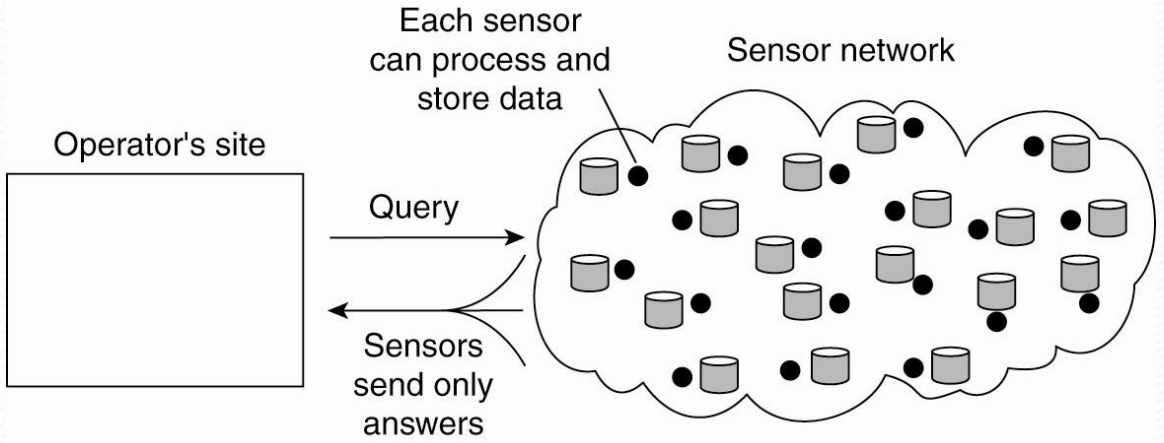
- ❑ ни одна машина не имеет полной информации о состоянии системы;
- ❑ машины принимают решения на основе только локальной информации;
- ❑ выход из строя одной машины не должен приводить к отказу алгоритма;
- ❑ не должно быть неявного предположения о существовании глобальных часов.

Пример. Мониторинг сердечного ритма



Пример. Мониторинг сердечного ритма





Основные темы

- Достоинства распределенных систем

  Прозрачность. Открытость. Масштабируемость.
- Процессы

  Процессы и потоки выполнения (нити). OpenMP. Многопоточные клиенты и серверы. Взаимное исключение критических интервалов. Алгоритмы Деккера, Петерсона. Семафоры Дейкстры. Механизм событий.

  Классические задачи взаимодействия процессов – «производитель-потребитель» и «читатели-писатели».
- Коммуникации

  Модели взаимодействия. Модель передачи сообщений. MPI. Режимы передачи сообщений. Коллективные операции. Удаленный вызов процедур (Remote Procedure Call).
- Синхронизация

  Синхронизация времени. Логические часы. Глобальное состояние. Алгоритмы голосования. Взаимное исключение. Распределенные транзакции. Координация процессов.

Основные темы

- Распределенная разделяемая память (DSM)
  Достоинства разделяемой памяти. Принципы реализации распределенной разделяемой памяти. Модели консистентности. Страничная DSM. DSM на базе разделяемых переменных.
- Распределенные файловые системы
  Доступ к директориям и файлам. Семантика одновременного доступа к одному файлу нескольких процессов. Кэширование и размножение файлов. Примеры - Network File System, HDFS.
- Отказоустойчивость
  Типы отказов. Поломка. Пропуск данных. Ошибка синхронизации. Ошибка отклика. Византийские ошибки. Надежная групповая рассылка. Протоколы двухфазного и трехфазного подтверждения. Фиксация контрольных точек и восстановление после отказа. Протоколирование сообщений.
- Примеры распределенных систем
  Проект Hadoop.

Вопросы по курсу

- В транспьютерной матрице размером 4*4, в каждом узле которой находится один процесс, необходимо переслать очень длинное сообщение (длиной L байт) из узла с координатами (0,0) в узел с координатами (3,3). Сколько времени потребуется для этого, если передача сообщений точка-точка выполняется в буферизуемом режиме MPI? А сколько времени потребуется при использовании синхронного режима и режима готовности? Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

- Все 16 процессов, находящихся на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания), одновременно выдали запрос на вход в критическую секцию. Сколько времени потребуется для прохождения всеми критических секций, если используется древовидный маркерный алгоритм (маркером владеет нулевой процесс). Время старта (время «разгона» после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

Вопросы по курсу

- Сколько времени потребует выбор координатора среди 16 процессов, находящихся в узлах транспьютерной матрицы размером 4*4, если используется круговой алгоритм? Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память считаются бесконечно быстрыми.

- Последовательная консистентность памяти и алгоритм ее реализации в DSM с полным размножением. Сколько времени потребует модификация 10 различных переменных 10-ю процессами (каждый процесс модифицирует одну переменную), находящимися на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания) и одновременно выдавшими запрос на модификацию. Время старта (время «разгона» после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

Вопросы по курсу

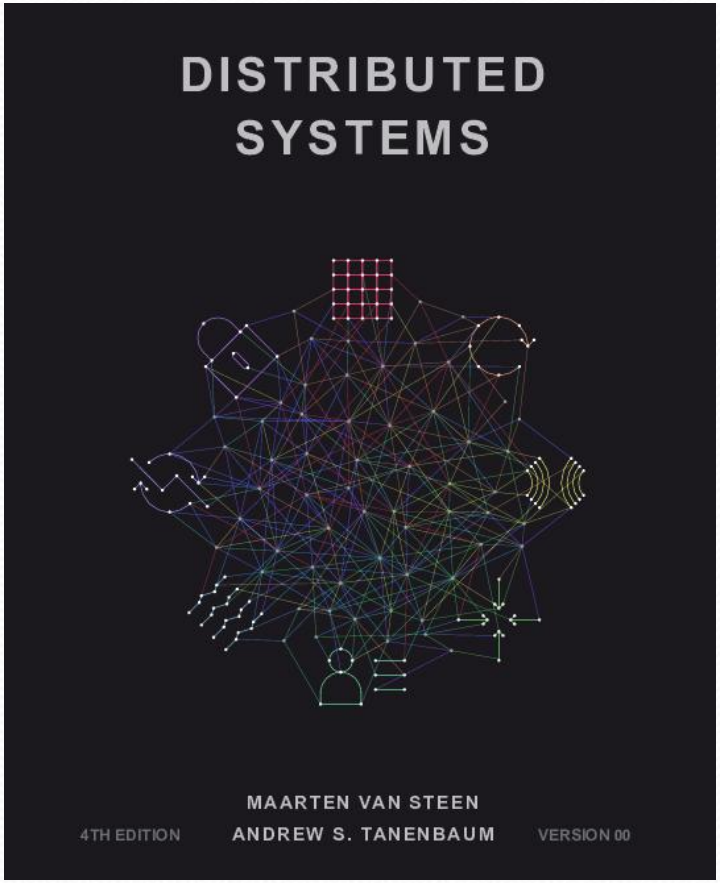
- Протоколы голосования. Алгоритмы и применение. Дайте оценку времени выполнения одним процессом 2-х операций записи и 10 операций чтения N байтов информации с файлом, расположенным (размноженным) на остальных 10 ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания). Определите оптимальные значения кворума чтения и кворума записи для N=300. Время старта (время «разгона» после получения доступа к шине для передачи) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса (при одновременных запросах - в порядке номеров ЭВМ). Операции с файлами и процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

- Алгоритм надежных и неделимых широковещательных рассылок сообщений. Дайте оценку времени выполнения одной операции рассылки для сети из 10 ЭВМ с шинной организацией (без аппаратных возможностей широковещания), если отправитель сломался после посылки 5-го сообщения. Время старта (время «разгона» после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

Литература

- Э. Таненбаум, М. ван Стеен. Распределенные системы. Принципы и парадигмы.– СПб.: Питер, 2003. — 877 с.: ил. — (Серия «Классика Computer Science») — ISBN 5–272–00053–6.
- В.А. Крюков, В.А. Бахтин. Распределенные системы. http://sp.cs.msu.su в разделе «Информация».
- Антонов А.С. Технологии параллельного программирования MPI и OpenMP: Учеб. пособие. Предисл.: В.А.Садовничий. - Серия «Суперкомпьютерное образование». М.: Издательство Московского университета, 2012.-344 с.
- Э. Таненбаум. Современные операционные системы. 3-е изд. - СПб.: Питер, 2010. — 1120 с. .: ил. — (Серия «Классика Computer Science») — ISBN 978-5-459-00757-2, 978-0136006633.

Литература



https://www.distributed-systems.net/

M. van Steen and A.S. Tanenbaum, Distributed Systems, 4th ed., distributed-systems.net, 2025