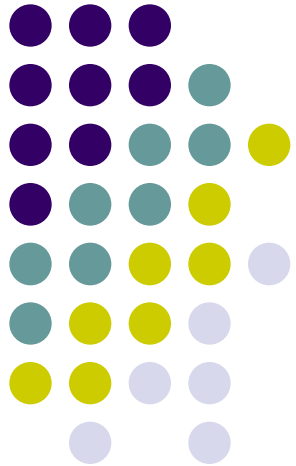
Распределенные системы



Лекция 2

Математическое представление распределенной системы

к.т.н. Приходько Т.А.

По материалам учебного пособия Микова А.И.
"Распределенные компьютерные системы и алгоритмы"

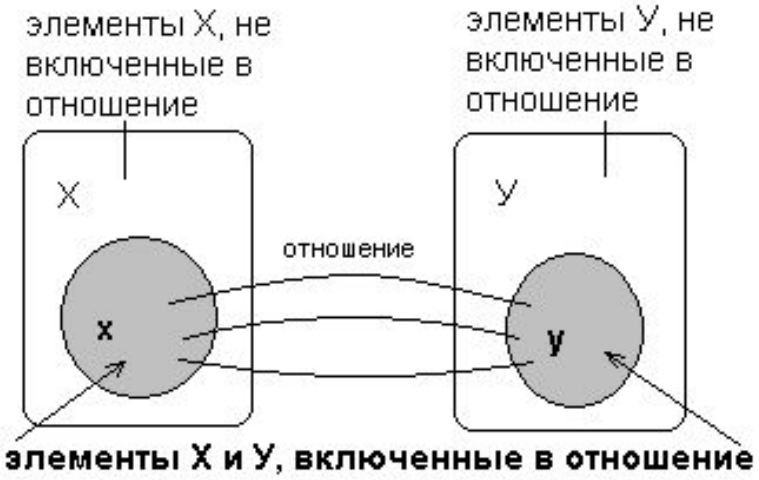
Лекция 2

- Математическое представление распределенной системы
- Сосредоточенные и распределенные системы
- Распределенные задачи и алгоритмы
- Надежность и безопасность распределенных систем

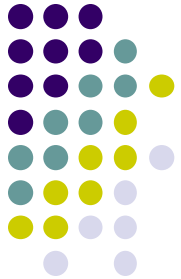
Математическое представление распределенной системы

- Под ***системой*** понимается множество элементов и связей между ними.
- Обозначим V – множество элементов системы. Тогда бинарное отношение $R_2 \subseteq V \times V$ задает наличие попарных связей между элементами. Если для некоторых элементов $x \in V$ и $y \in V$ пара $(x, y) \in R_2$, то в системе существует связь от x к y. Если $(x, y) \notin R_2$, то такой связи нет.

- Порядок элементов в паре важен, так как связи могут быть направленными, несимметричными.



Математическое представление распределенной системы



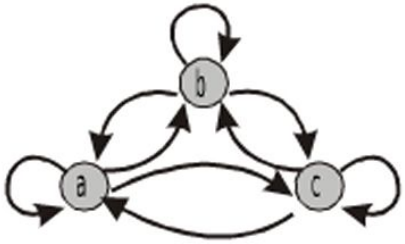
- В системе могут быть не только попарные связи, но и связи троек элементов. Такие связи описываются тернарными отношениями $R_3 \subseteq V \times V \times V = V^3$. Например, связь «ребенок и его родители» – связь трех элементов.
- В общем случае в системе могут быть также связи, задаваемые отношениями $R_4 \subseteq V^4$, $R_5 \subseteq V^5$.,…, $R_n \subseteq V^n$. Здесь n – количество элементов в системе.

С каждым отношением связан определенный смысл, выражаемый высказыванием, например, «*x* следует за *y*», «*x* посылает сигнал *y*», «*x* – ребенок *y* и *z*» и т.д. Иначе говоря, вместо отношений (или вместе с отношениями) удобно рассматривать соответствующие ***предикаты*** P_2 , P_3 , P_4 ,…, P_n . В дополнение к перечисленным рассматривают и предикаты P_1, которые можно интерпретировать как выражение свойств элементов множества V.

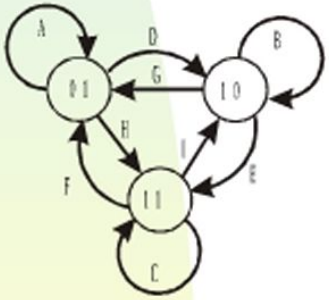
Бинарное отношение

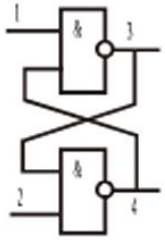
Бинарное отношение эквивалентности

- Обозначение: $R_{\sim}$
- Граф



- Рефлексивность: $x \sim x$
- Симметричность: $x \sim y \Leftrightarrow y \sim x$
- Транзитивность: $x \sim y,\ y \sim z \Rightarrow x \sim z$
- **Пример**





Бинарное отношение эквивалентности $R_{\sim}$

=

Рефлексивность

+

Симметричность

+

Транзитивность

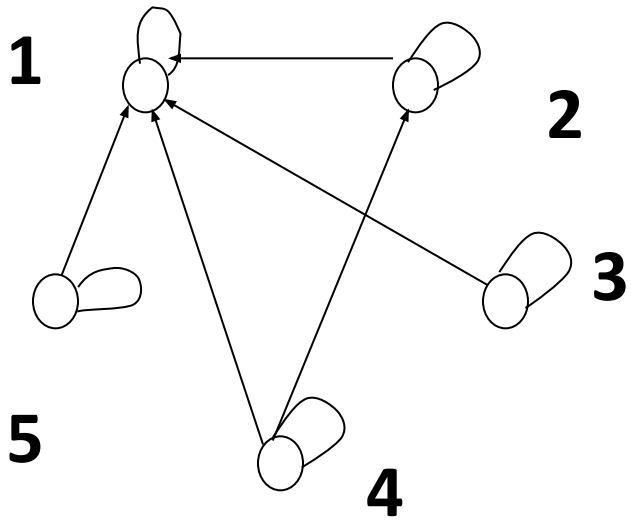
Бинарное отношение

- ***Пример***

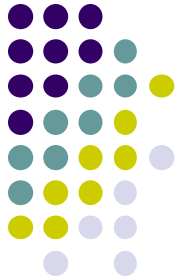
Матрица смежности:

	1	2	3	4	5
1	1	0	0	0	0
2	1	1	0	0	0
3	1	0	1	0	0
4	1	1	0	1	0
5	1	0	0	0	1

Графическое задание:



Математическое представление распределенной системы



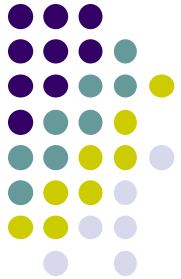
- Подчеркнем, что бинарных отношений в системе может быть несколько. Например, в цилиндре двигателя автомобиля газ (бензино-воздушная смесь), загораясь, толкает поршень и, одновременно, нагревает его. Т.е. существует отношение ***«x толкает y»*** и отношение ***«x нагревает y».*** Ясно, что в зависимости от целей исследования системы одни отношения рассматриваются как существенные, а другие – как второстепенные.
- В общем случае систему можно описать как набор

$$S = \{V, \{P_{i,j}\}\},$$

где

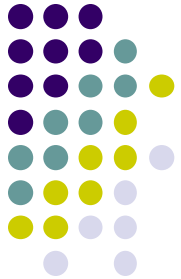
  - индекс ***i*** обозначает арность отношения (или количество мест предиката),
  - индекс ***j*** дает возможность различать отношения одной и той же арности.

Математическое представление распределенной системы



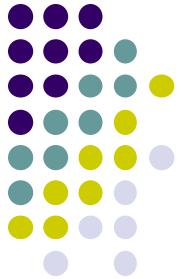
- Некоторые из предикатов $P_{1,j}$ могут характеризовать ***местоположение элемента системы***. Например, его географические координаты, пространственные координаты (спутник связи), нахождение в определенном помещении (компьютер локальной сети) и т.д. **Подмножества элементов, имеющих одно и то же (в пределах некоторого допуска, приближения) местоположение, мы будем называть *сайтами*.**
- Аналогично, некоторые из предикатов $P_{2,j}$ могут характеризовать ***взаимное расположение элементов***, например, расстояние, время передачи сигнала, стоимость переноса информации или вещества от одного элемента системы к другому. Взаимное расположение может быть существенным и для троек элементов, и для четверок и т.д., и выражаться соответствующими предикатами.

Математическое представление распределенной системы



- ***Распределенными системами*** будем называть такие системы, для которых предикаты местоположения элементов или групп элементов играют существенную роль с точки зрения функционирования системы, а, следовательно, и с точки зрения анализа и синтеза системы.
- Распределенные системы могут быть **непрерывными и дискретными**.
  - Непрерывные распределенные системы характеризуются бесконечным количеством элементов;
  - Элементы дискретной системы четко «очерчены», отделены друг от друга. Один из видов отношений – бинарное отношение «быть соседними элементами».

Сосредоточенные и распределенные системы



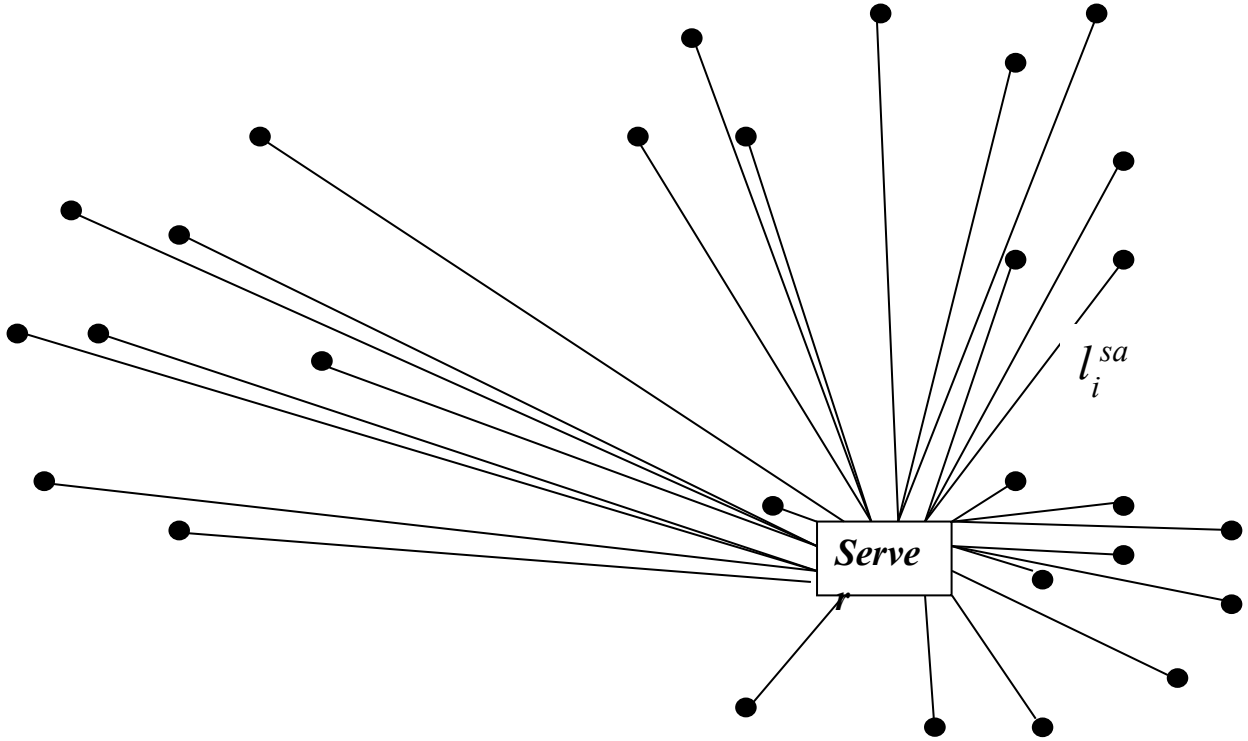
- Обозначим две такие системы *Sd* и *Ssa* (от английских терминов distributed и stand-alone), тогда множество элементов в каждой из систем, обычно, можно разделить на два подмножества:
  - ***U*** – подмножество сосредоточенных элементов, занимающих относительно небольшой объем пространства и реализующих некоторую функцию преобразования.
  - ***W*** - подмножество состоящее из элементов связей элементов ***U*** между собой. Их основная задача не преобразование, а передача чего-либо в системе от одного элемента к другому.

Тогда формальное описание систем Sd и Ssa:

$$V(Sd) = Ud \cup Wd$$

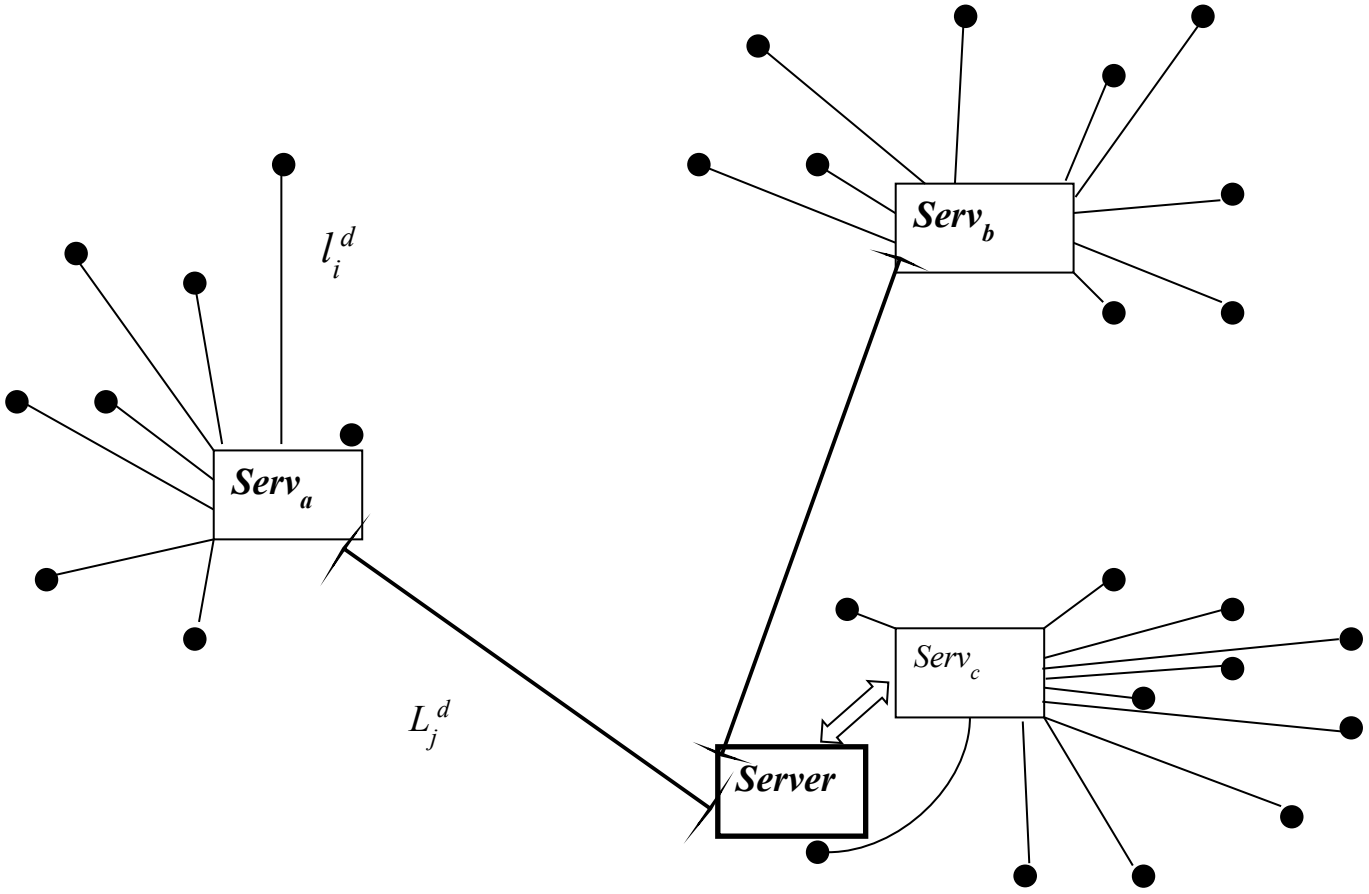
$$V(Ssa) = Usa \cup Wsa$$

Сосредоточенные и распределенные системы



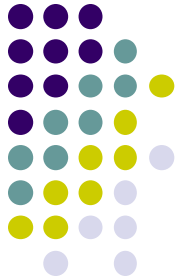
Сосредоточенная система

Сосредоточенные и распределенные системы



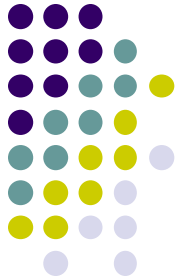
Распределенная система

Сосредоточенные и распределенные системы



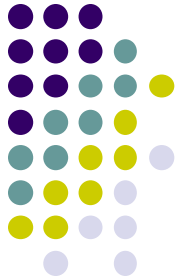
- элементы из множества ***Wd (связи распред. систем)*** могут быть весьма разнообразными, с большим количеством характеристик. Примером является передача сигналов посредством радиосвязи на сверхвысоких частотах. При этом эфир становится элементом системы. Поскольку радиоволны распространяются во всех направлениях, то эфир характеризуется тремя пространственными координатами *x*, *y* и *z*.
- Напротив, в **сосредоточенных системах** элементы из множества ***Wsa*** описываются просто, а часто вообще исключаются из рассмотрения как несущественные для анализа свойств системы. Таким элементом, например, в компьютере является набор проводников, соединяющих процессор с микросхемами памяти.

Сосредоточенные и распределенные системы



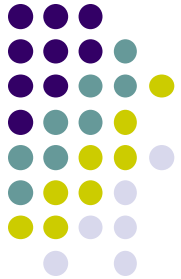
- Множества элементов ***Usa*** и ***Ud*** сосредоточенной и распределенной систем также могут отличаться.
- В сосредоточенной системе функционирование элементов из ***Usa*** инвариантно их местоположению.
- В распределенной системе функционирование элементов из ***Ud*** в общем случае зависит от их местоположения. Эта зависимость может быть нескольких видов:
  - 1) зависимость от **источников информации**, имеющих определенное местоположение;
  - 2) зависимость от **поставленных задач**, которые должны решаться элементами системы;
  - 3) зависимость от **параметров среды** в различных точках.

Тандемы распределенных систем



- Рассмотрим две системы, **S1** и ***S2***.
  - Первая система функционирует для достижения некоторой цели **G1**. При этом в любой момент времени имеется некоторая степень достижения этой цели.
  - Вторая система функционирует для того, чтобы ускорить достижение цели первой системой или увеличить степень достижения цели первой системой. Таким образом, система **S1** является основной, а система **S2** – вспомогательной.
  - Цель **G2** создания системы **S2** и/или цель ее функционирования является производной от цели **G1**.
- Таким образом, системы **S1** и **S2** образуют своего рода ***«тандем»,*** являющийся новой системой – фирмой с корпоративной информационной системой.

Тандемы DS



- Систему $S1$ можно описать как набор

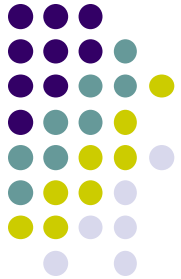
  $S_1 = \{V_1, \{P_{i,j}\}\}$, где

  - индекс i обозначает арность отношения (или количество мест предиката),
  - индекс j дает возможность различать отношения одной и той же арности.
  - Отдельные предикаты $P_{1,j}$ характеризуют местоположение элементов системы.
  - Некоторые из предикатов $P_{2,j}$ характеризуют взаимное расположение элементов.
- Соответственно, система S_2 описывается как набор

  $S_2 = \{V_2, \{Q_{i,j}\}\}$.
  - Здесь отдельные предикаты $Q_{1,j}$ характеризуют местоположение элементов системы,
  - отдельные предикаты $Q_{2,j}$ характеризуют взаимное расположение элементов системы S_2 .

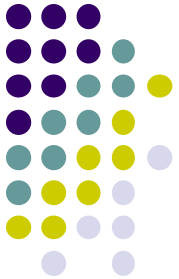
Множество элементов V_2 «порождается» множеством элементов V_1 . Множества предикатов $Q_{1,j}$ и $Q_{2,j}$ «зависят» от множеств предикатов $P_{1,j}$ и $P_{2,j}$. В частности, сайты системы S_2 формируются, как правило, на основе сайтов системы S_1 .

Распределенные задачи и алгоритмы



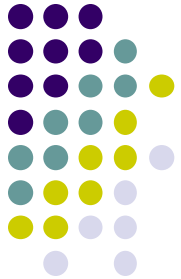
- Распределенная система порождает ***распределенную задачу***, поскольку исходные данные для задачи возникают в различных точках пространства. Также в различных точках требуются те или иные результаты решения задачи. Зачастую, задачу можно разбить на совокупность подзадач, некоторые из них могут быть сосредоточенными: все исходные данные возникают в одной точке пространства, и там же требуются результаты решения подзадачи.
- Распределенная задача решается ***распределенным алгоритмом***, собирающим исходные данные из разных точек и посылающим результаты расчетов в разные точки.

Распределенные задачи и алгоритмы



- Один из видов распределенных алгоритмов – ***протоколы***. Протокол характеризуется тем, что имеется как минимум две стороны, разделенные каналом связи. На каждой из сторон выполняется локальный (сосредоточенный) алгоритм A_k .
- Локальный алгоритм A_1 выполняется до некоторого момента времени, когда для продолжения работы ему требуются данные от другого локального алгоритма A_2 . Он посылает через линию связи запрос на данные локальному алгоритму A_2 . Алгоритм A_2 отвечает, пересылая сообщение по линии связи. После этого локальный алгоритм A_1 продолжает свою работу.
- Протоколы, обычно, играют техническую роль и служат для установления режимов приема/передачи данных между удаленными объектами. В вычислительном отношении локальные алгоритмы – части протокола – не являются сложными.

Распределенные задачи и алгоритмы



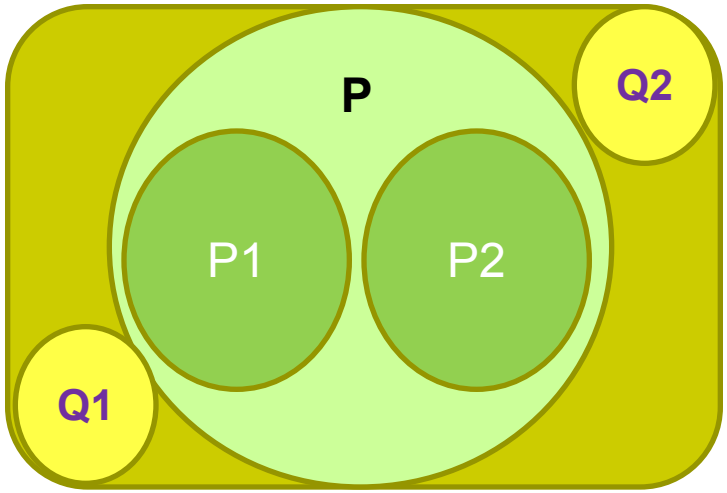
Криптографические протоколы

- Особый вид распределенных алгоритмов – ***криптографические протоколы.*** Они предназначены для доказательства одной или обеими сторонами, что они именно те, за кого себя выдают, для обмена конфиденциальной информацией и других подобных целей.
- Рассмотрим один из простейших криптографических протоколов – распределенных алгоритмов.

Распределенные задачи и алгоритмы

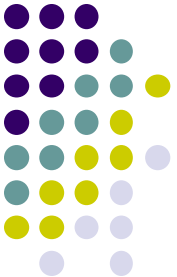
Криптографические протоколы

- Предположим, что в разных точках пространства находятся два объекта, O_1 и O_2 , каждый из которых решает свою часть (P_1 и P_2) общей задачи P. Вместе с тем, кроме стремления решить общую задачу, объекты имеют и свои частные задачи, Q_1 и Q_2 .
- Таким образом, объект O_1 решает одновременно задачи P_1 и Q_1, а объект O_2 – задачи P_2 и Q_2 . И, если задачи P_1 и P_2 совместимы и взаимно дополняют друг друга, то задачи Q_1 и Q_2 – противоречат друг другу.
- Следовательно, объекты O_1 и O_2 сотрудничают, но, одновременно, и конкурируют. А можно сказать и, что объекты конкурируют, но вынуждены сотрудничать. Смотря как расставить акценты, какая из задач, P_i или Q_i , важнее для объекта O_i .



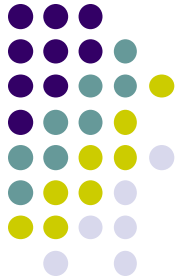
Распределенные задачи и алгоритмы

Криптографические протоколы



- В нашей задаче объекты O_1 и O_2 должны прийти к общему компромиссному решению в интересах решения задачи P.
- Предположим, что имеется два равноценных (с точки зрения задачи P) решения. Одно из них также устраивает объект O_1, а другое – устраивает объект O_2. Тогда объекты O_1 и O_2 решают бросить жребий, однако, бросить жребий и «увидеть» результат, находясь на большом расстоянии – весьма затруднительно.
- Процедуру мог бы выполнить объект O_1 и сообщить результат объекту O_2. Но объект O_2 не вполне доверяет объекту O_1.
- Выход состоит в следующем:

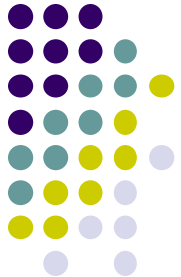
Распределенные задачи и алгоритмы



Криптографические протоколы

- Обозначим одно из решений числом ***0***, другое – числом ***1***. Каждый из объектов независимо от другого должен назвать какое-нибудь целое число n_i в пределах, например, от ***0*** до ***99***. Затем объекты обмениваются числами n_i и вычисляют результат $(n_1 + n_2) \pmod 2$. Это и будет решение, т.е. число ***0*** или ***1***.
- Абсолютно одновременно переслать друг другу числа n_i объекты не могут. Кто-то пришлет свое число первым и окажется в невыигрышном положении. Например, O_1 переслал свое число n_1 объекту O_2, заинтересованному в том, чтобы выиграло решение «1». Тогда O_2, зная n_1, решает уравнение $(n_1 + n_2) \pmod 2 = 1$ относительно переменной n_2, и посылает O_1 найденное значение n_2. Решение уравнения практически не требует времени: получив от O_1 четное число, O_2 должен ответить нечетным, и наоборот.

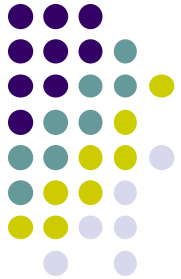
Распределенные задачи и алгоритмы



Криптографические протоколы

- Эта несправедливость должна быть устранена. Ясно, что какая-то из сторон обмена сообщениями первой пришлет свое число.
- Необходимо сделать так, чтобы у второй стороны не хватило времени на «подбор ответа».
- Выход:
  - Объекты O_1 и O_2 могут договориться, что вся процедура «бросания жребия» на расстоянии должна завершиться за несколько минут. Если, при этом, оба объекта знают, что ***подбор ответа*** требует нескольких часов работы суперЭВМ, то они могут быть спокойны за то, что решение не «подстроено» другой стороной.
  - Для обеспечения таких временн***ы***х параметров в вычислениях должна использоваться функция $f(x)$, значение которой $y = f(x)$ при известном аргументе x вычислить можно относительно быстро. Но вот решить уравнение $f(y) = x$, т.е. отыскать неизвестное x при известном значении y быстро нельзя. Более того, желательно, чтобы не было известно никаких математических методов для решения этого уравнения, кроме полного перебора или подобного ему по сложности.

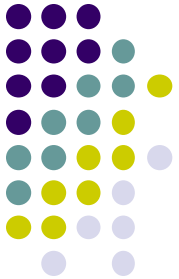
Распределенные задачи и алгоритмы



Криптографические протоколы

- Распределенный алгоритм бросания жребия состоит из шагов:
  - 1. Объект O_2 выбирает случайным образом число x из большого интервала $[0, q-1]$. Вычисляет $y = f(x)$.
  - 2. Объект O_2 пересылает число y объекту O_1 . Объект O_1 не сможет восстановить число x.
  - 3. Объект O_1 выбирает случайным образом число z из большого интервала $[0, q-1]$, и случайный бит w. Эти действия могут выполняться одновременно с п.1.
  - 4. Объект O_1 вычисляет $s = h(z, y, w)$. Число z здесь необходимо для «маскировки» бита w, а число y – для «проверки» объектом O_2 правильности действий объекта O_1 .
  - 5. Объект O_1 пересылает число s объекту O_2 . Бит w отправляется объекту O_2 , но он «запрятан» в числе s. Объект O_2 не сможет восстановить этот бит. Объект O_2 не сможет восстановить и число z.
  - 6. Объект O_2 выбирает случайный бит c и отправляет *его* объекту O_1 открытой пересылкой.
  - 7. Объект O_1 пересылает число z и бит w объекту O_2 . Открытая пересылка. Объект O_1 уже может определить результат бросания жребия: $c + w \pmod 2$.
  - 8. Объект O_2 вычисляет $t = h(z, y, w)$. Здесь z и w только что получены от O_1 , а y было вычислено в п.1.
  - 9. Объект O_2 сравнивает t и s, ранее полученное от объекта O_1 .
  - 10. Если $t = s$, то объект O_2 вычисляет результат бросания жребия: $c + w \pmod 2$.

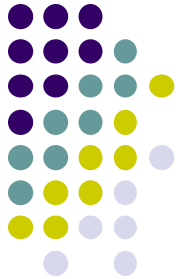
Распределенные задачи и алгоритмы



Криптографические протоколы (дополнительные условия)

- Целые числа, с которыми приходится оперировать, должны иметь в десятичной записи не менее 150-200 цифр или не менее 512 бит в двоичной записи. Такие числа называют «длинными».
- Пусть $f(x)$ и $h(z, y, w)$ – две таких трудно обращаемых функции.
  - В функции $h(z, y, w)$ первые два аргумента – длинные числа, а третий – битовый.
  - Функции f и h известны объектам O_1 и O_2 , и они владеют алгоритмами быстрого вычисления значений этих функций при заданных значениях аргументов, но не умеют быстро обращать эти функции, т.е. решать уравнения.

Распределенные задачи и алгоритмы

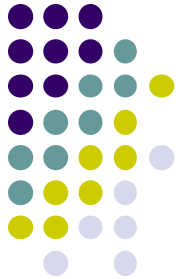


- Функции *f* и *h* могут быть различными. В частности, используются функции:

$$f(x) = g^x \pmod p \quad \text{и} \quad h(z, y, w) = y^w g^z \pmod p.$$

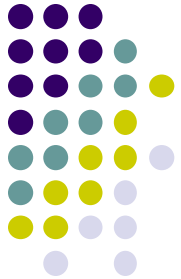
- Здесь «секретные» значения *x*, *w*, *z* находятся в показателях степеней, и для того, чтобы найти их, требуется решить задачу дискретного логарифмирования, для которой эффективный алгоритм, существенно лучший полного перебора, неизвестен.
- Константы *p*, *q*, *g* должны быть известны тому и другому объектам. Число *p* – длинное простое число. Число *q* – также длинное простое число, являющееся делителем числа *p* – 1. Число $g < p$, не равное 1, удовлетворяет условию

$$g^q = \pmod p.$$



Надежность и безопасность распределенных систем

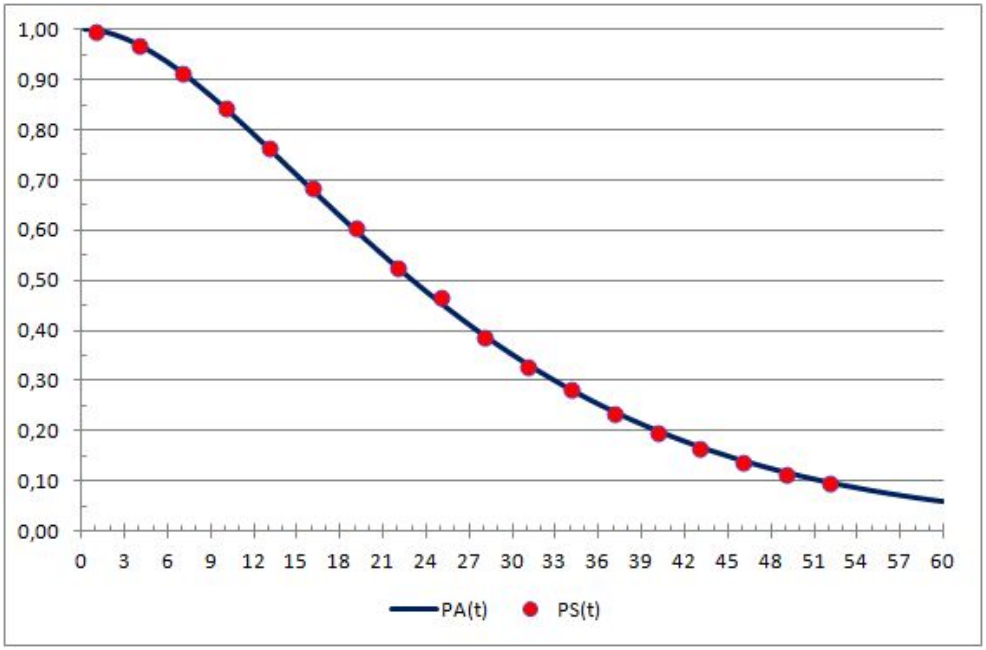
Надежность и безопасность распределенных систем



- Под ***надежностью*** понимается в соответствии с ГОСТ 27.002-89 *свойство системы сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания и транспортирования.*
  - Многие системы не являются абсолютно надежными, т.е. свойство надежности системы проявляется на *конечном интервале времени эксплуатации системы*, по истечении которого происходит ***отказ*** в работе. Длительность интервала безотказной работы зависит от очень большого числа факторов, предсказать которые нереально, поэтому, отказ обычно считают ***случайным событием***.

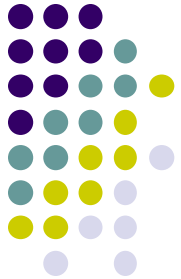
Надежность и безопасность распределенных систем

- Надежность принято характеризовать ***вероятностью отказа*** в работе (или ***вероятностью безотказной работы***) в течение определенного отрезка времени.
- Другой характеристикой надежности системы является ***среднее время наработки на отказ***.



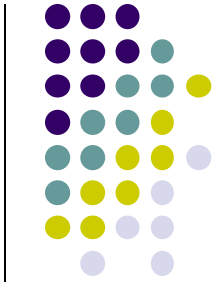
Функция **надежности** тестовой МАС PA(t) и экспериментальные значения вероятности безотказной работы PS(t)

Надежность и безопасность распределенных систем



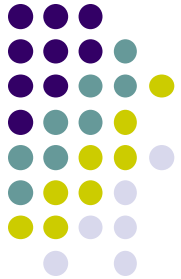
- Под ***безопасностью*** понимается состояние *защищенности системы от потенциально и реально существующих угроз, или отсутствие таких угроз.* Система находится в состоянии безопасности, если действие внешних и внутренних факторов не приводит к ухудшению или невозможности ее функционирования.
- Угрозы могут быть различного рода, в том числе угроза физического разрушения.
- В контексте нашей темы интересны угрозы ***информационные***. К ним относятся угрозы получения системой
  - недостоверной входной информации,
  - искажения внутрисистемной информации,
  - утечка информации о функционировании системы.

Надежность и безопасность распределенных систем



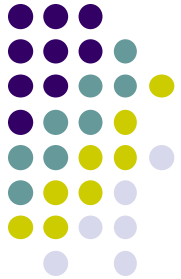
- ***Информационная безопасность*** — состояние защищенности информационной среды общества, обеспечивающее ее формирование, использование и развитие в интересах граждан, организаций, государства.
- В качестве стандартной модели безопасности часто приводят модель CIA:
  - ***конфиденциальность*** информации – confidentiality (обязательное для выполнения лицом, получившим доступ к определенной информации, требование не передавать такую информацию третьим лицам без согласия ее владельца);
  - ***целостность*** (integrity) - гарантия существования информации в непротиворечивом виде;
  - ***доступность*** (availability) - возможность получение информации авторизованным пользователем в нужное для него время.

Надежность и безопасность распределенных систем



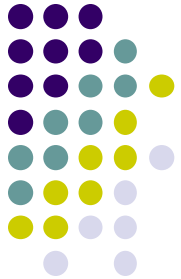
- Выделяют и другие категории безопасности:
  - ***аутентичность*** — возможность установления автора информации;
  - ***апеллируемость*** — возможность доказать что автором является именно заявленный человек, и не никто другой.
- Все физические элементы любой системы являются потенциально ненадежными и уязвимыми с точки зрения безопасности.

Надежность и безопасность распределенных систем



- Ненадежность элементов системы, осуществляющих переработку информации, может заключаться
  - в полном отказе от переработки,
  - в изменении функции (стабильном получении неверных результатов),
  - в сбоях (периодическом возникновении ошибок).
  - в полном прекращении передачи, в одностороннем прекращении передачи (для двунаправленных каналов),
  - в возникновении случайных ошибок при передаче (помех).

Надежность и безопасность распределенных систем

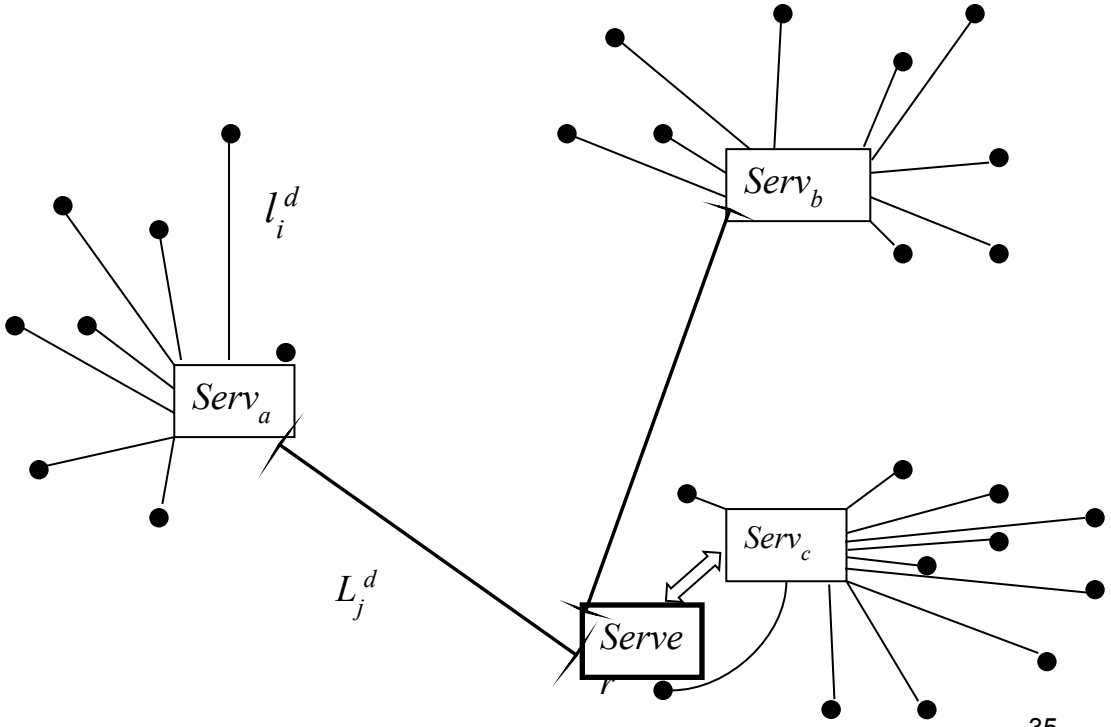


- Таким образом, проблемы надежности и безопасности во многом родственны. Они связаны с вмешательством в функционирование системы. Различие заключается в том, что ненадежность определяется физическими, природными факторами и не связана с чьими-то целями.
- Небезопасность определяется, в основном, «человеческим фактором» - наличием злоумышленников и/или беспечных сотрудников. Но одна из проблем безопасности – утечка информации при несанкциони-рованном доступе – не имеет аналога среди проблем надежности.

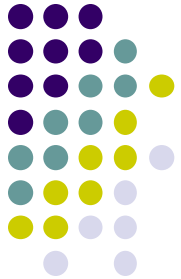
Надежность и безопасность распределенных систем

- В распределенной системе количество элементов больше, чем в сосредоточенной: S_d включает дополнительные серверы $Serv_a$, $Serv_b$, $Serv_c$ и дополнительные элементы (линии связи) L_j . Количество линий связи l_i объектов с серверами в сосредоточенной и распределенной системах одинаково – оно определяется количеством объектов.

- Каждый фактор с точки зрения надежности, если его рассматривать изолированно, играет положительную или отрицательную роль.

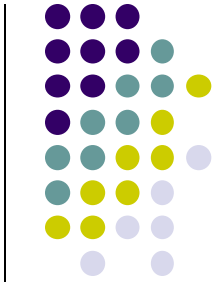


Надежность и безопасность распределенных систем



- ***Негативно влияет на надежность***:
  - увеличение количества ненадежных элементов в системе при прочих равных условиях играет отрицательную роль. Под прочими равными условиями здесь понимается
    - неизменность архитектуры (соединений и распределения функций) системы,
    - неизменность параметров элементов и проч.
  - Если же архитектуру изменить, например, использовать дополнительные элементы для дублирования (резервирования), то надежность, напротив, повышается.
  - Увеличение длины линий связи: Обычно с увеличением длины линии увеличивается
    - количество помех,
    - стоимость передачи,
    - возможности злоумышленников по съему информации или по ее искажению.

Методики построения алгоритмов для обеспечения надежности DS

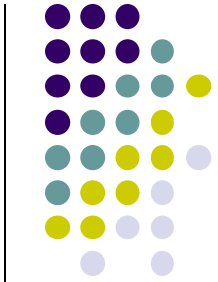


Все эти факторы надо учитывать при разработке ***методики построения алгоритмов***:

- Она должна быть рассчитана на возможные сбои или рассогласования в работе узлов системы (при программировании в сосредоточенных системах возможность сбоя обычно не учитывается).

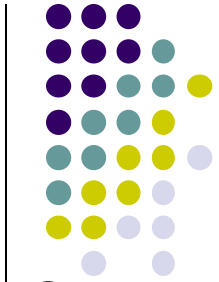
- В настоящее время существует два подхода к разработке распределенных алгоритмов:
  - построение отказоустойчивых алгоритмов и
  - построение стабилизирующих алгоритмов.
- В ***устойчивых*** алгоритмах каждый шаг каждого процесса предпринимается с достаточной осторожностью, чтобы гарантировать, что, несмотря на сбои, правильные процессы выполняют только правильные шаги.
- В ***стабилизирующих*** алгоритмах правильные процессы могу быть подвержены сбоям, но алгоритм в целом гарантирует исправление ошибок.

Устойчивые алгоритмы



- Разработаны так, чтобы учитывать возможность сбоев в некоторых процессах (относительно небольшом их количестве) и гарантировать при этом правильность выполнения тех процессов, в которых не произошло сбоев.
- Эти алгоритмы используют такие стратегии как голосование, вследствие чего процесс воспримет только такую информацию извне, о получении которой объявит достаточно много других процессов. Однако процесс никогда не должен ждать получения информации от всех процессов, потому что может возникнуть тупик, если при выполнении какого-либо процесса произойдет сбой.
- Устойчивые алгоритмы защищают систему от *отказов* ограниченного числа узлов. Остающиеся работоспособными узлы поддерживают правильное (хотя возможно менее эффективное) поведение во время восстановления и реконфигурации системы.
- Следовательно, устойчивые алгоритмы должны использоваться, когда невозможно временное прерывание работы.

Стабилизирующие алгоритмы

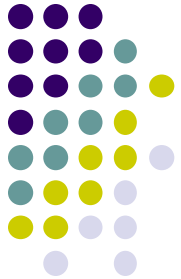


- Предлагают защиту против *временных сбоев*, то есть, временного аномального поведения компонент системы. Эти сбои могут происходить в больших частях распределенной системы, когда физические условия временно достигают критических значений, стимулируя ошибочное поведение памяти и процессоров.
- Примером может быть система управления космической станцией, когда станция подвергается сильному космическому излучению, а также системы, в которых на многие компоненты одновременно воздействуют неблагоприятные природные условия. Когда воздействие этих условий исчезает, процессы восстанавливают свою работоспособность и функционируют на основе программ. Однако из-за их временного аномального поведения глобальное результирующее состояние системы может быть непредвиденным. Свойство стабилизации гарантирует сходимость к требуемому поведению.

Перечень лабораторных работ

№	Название
1.	Дополнить код релевантными командами MPI (четные отправляют, нечетные принимают сообщения) (отчет не требуется)
2.	Тема: «Двухточечные обмены». Передача ранга и суммы рангов по кольцу в блокирующем и неблокирующем режимах.
3.	Тема: «Двухточечные обмены». Задача фильтрации+Waitall().
4.	Тема: «Двухточечные обмены». Программа с пробниками.
5.1 5.2	Тема: «Двухточечные и коллективные обмены». Выполнить перемножение векторов в режимах **двухточечного и коллективного обмена**. Рассмотреть зависимость времени исполнения программы от количества процессов и размерности векторов для блокирующей, неблокирующей, по готовности и в режиме коллективного обмена master-slave.
6.	Выполнить задачу фильтрации в коллективном исполнении.
7-8.	Тема: «Коллективные обмены». Задачи с графами.
9.	Тема: «Распределенная задача». Выполнить задачу обмена сообщениями на основе любого из изученных принципов middleware.

Математическое представление распределенной системы



Вопросы к экзамену:

1. Привести математическое описание распределенной системы (DS)
2. Охарактеризовать способ представления распределенных систем в виде графов.
3. Дать сравнительную характеристику сосредоточенных и распределенных систем
4. Понятие тандемов распределенных систем
5. В чем особенности Распределенных задач и алгоритмов?
6. В общих чертах охарактеризовать сложности разработки распределенных криптографических алгоритмов.
7. Понятия надежности и безопасность распределенных систем.
8. Методики построения алгоритмов для обеспечения надежности DS
9. Разновидности алгоритмов для обеспечения надежности DS