Лекция 03
РИС, III курс, ИСиТ+ПОИТ

Глобальное состояние распределенной системы

1. **Мультиагентные системы:** ERP, Глобальные электрические системы (электричество нельзя накопить, электростанцию нельзя остановить).

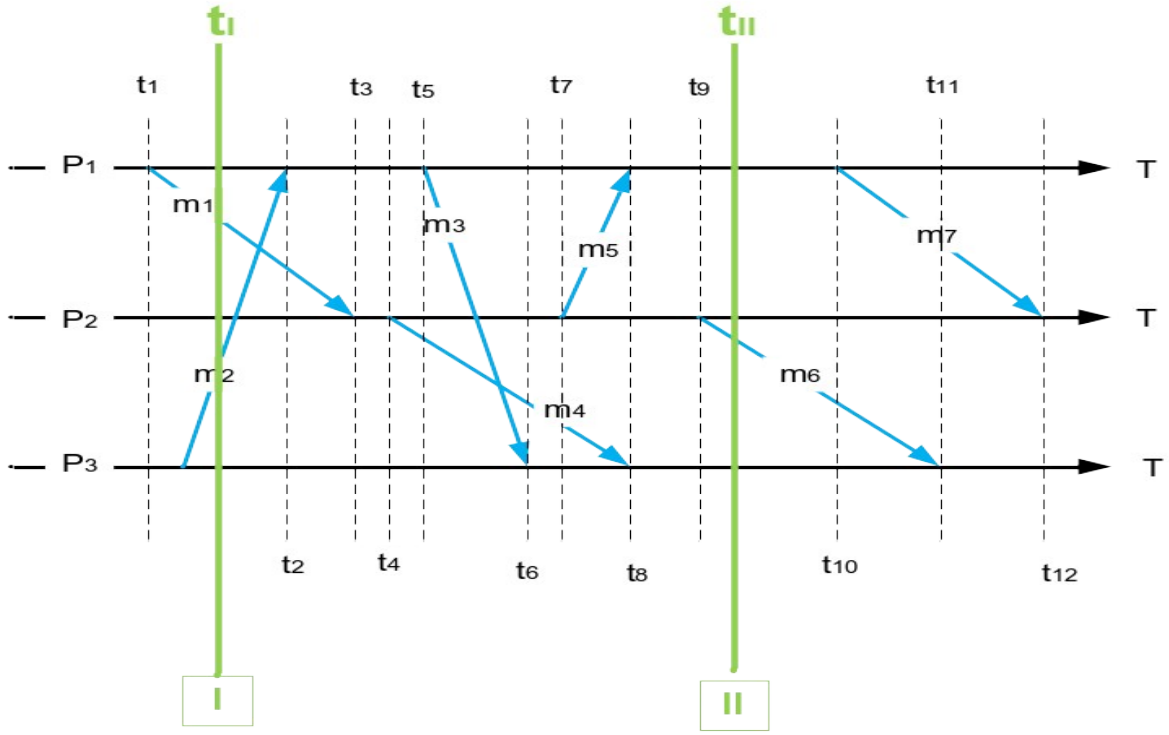
2. **Необходимость фиксации состояния:** с целью восстановления работоспособности, сбор статистики, проверка работоспособности.

3. **Постановка задачи:** определить, что значит согласованное состояние, сформулировать алгоритм получения глобального состояния распределенной информационной системы.

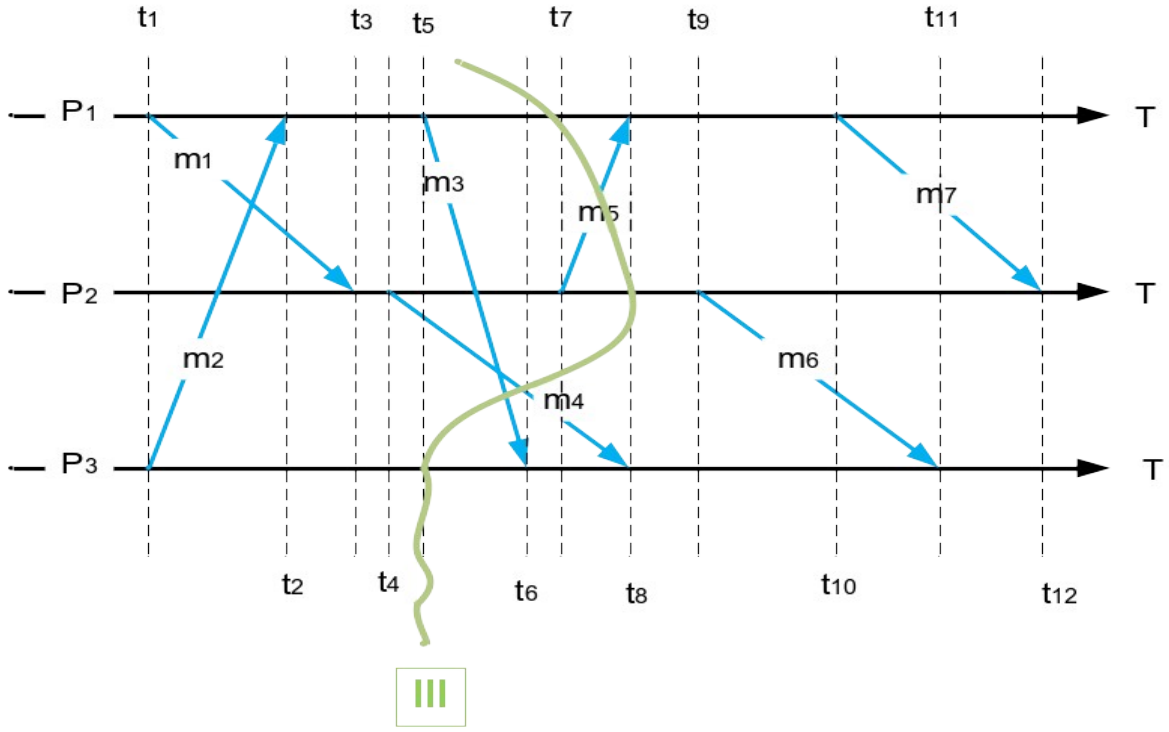
4. **Глобальное состояние распределенной системы:** совокупность ***согласованных*** локальных состояний всех ее компонент. В общем случае глобальное состояние зависит от решаемой задачи. В любом случае, каждый компонент распределенной системы принимает и/или отправляет сообщения, событиями каждого компонента можно считать прием и отправку сообщений.

5. **Глобальное состояние распределенной системы:** возможность восстанавливать работоспособность системы без потери данных.

6. **Модель глобального состояния распределенной системы:**

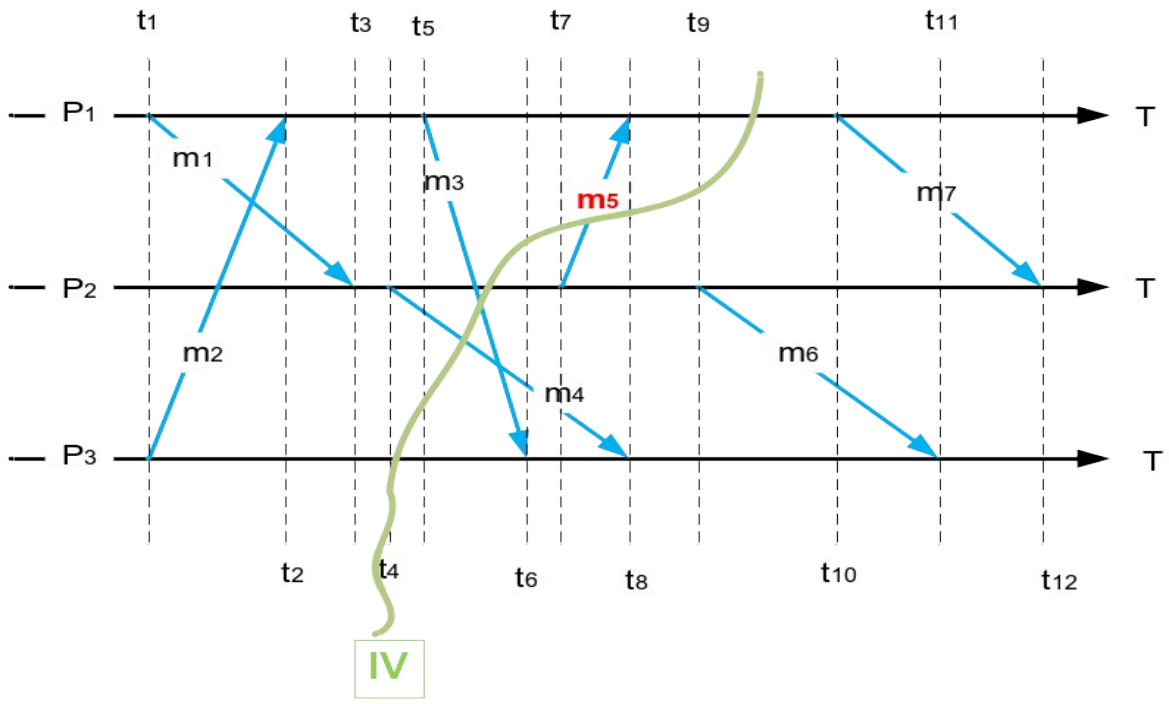



```
I:  P1[send:{m1}, receive:{}],
    P2[send:{}, receive:{}],
    P3[send:{m2}, receive:{}]
II: P1[send:{m1, m3}, receive:{m2, m5}],
    P2[send:{m4, m5, m6}, receive:{m1}],
    P3[send:{m2}, receive:{m3, m4}]
```




```
III: P1[send:{m1,m3},  receive:{m2}],
```

P2[send:{m4, m5}, receive:{m1}],
P3[send:{m2}, receive:{}]



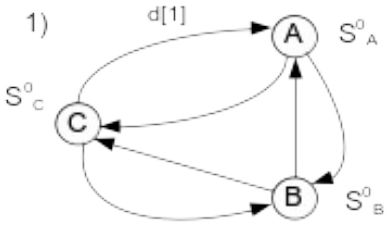
IV: P1[send:{m1,m3}, receive:{**m5**}],
P2[send:{m4}, receive:{m1}],
P3[send:{m2}, receive:{}]

7. **Согласованное локальное состояние:** нет полученных, но не отправленных сообщений.
8. **Согласованное состояние БД в ACID.**
9. **Алгоритм формирования глобального состояния:** все узлы связаны однонаправленными каналами, по которым они могут отправлять сообщения.
10. **Сообщение: D[K,P]** – информация (P) пересылаемая между узлами, имеет уникальный идентификатор (K).
11. **Инициатор:** узел, запрашивающий глобальное состояние.
12. **Маркер:** служебное сообщение **M[S,V,I]** – запрос на получение глобального состояния, содержащее уникальные идентификаторы состояния (S) и инициатора (I) и версию состояния(V).

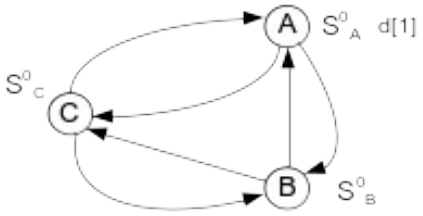
D[K]- сообщение, K-уникальный ключ

M[S, I]- маркер, S-состояние, I-инициатор

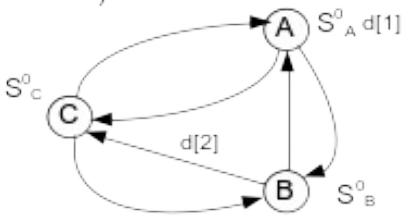
1)



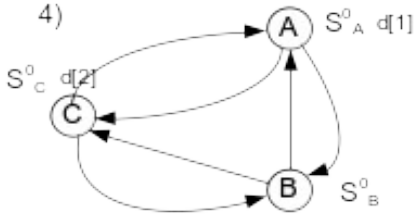
2)



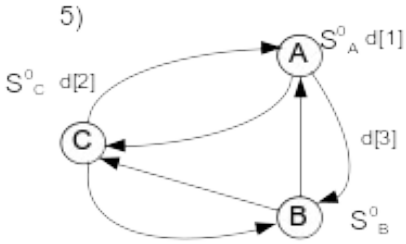
3)



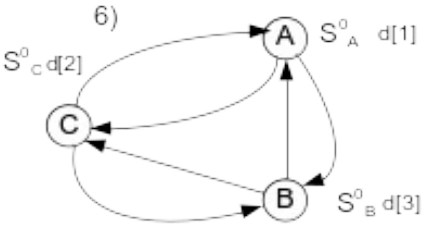
4)



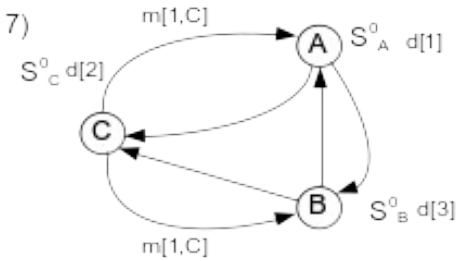
5)



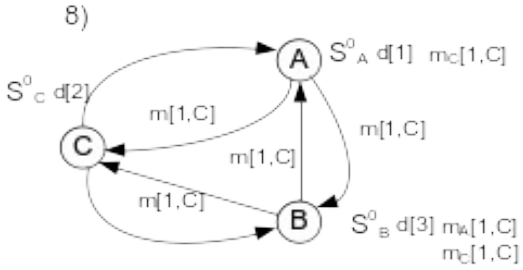
6)



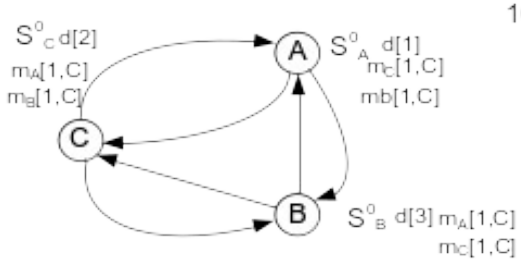
7)



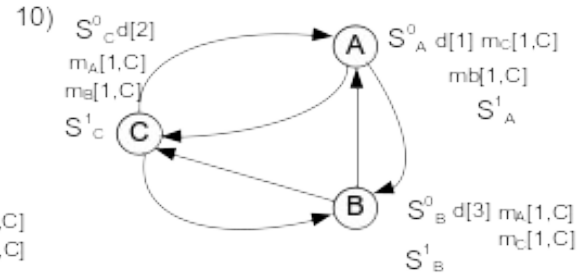
8)



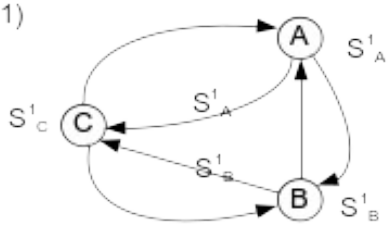
9)



10)



11)

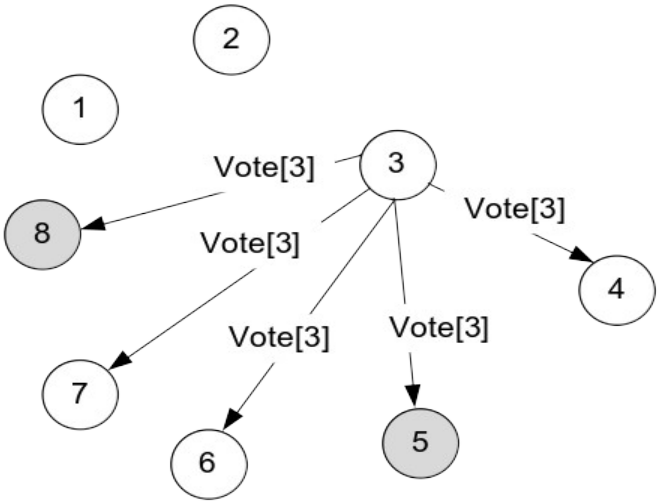


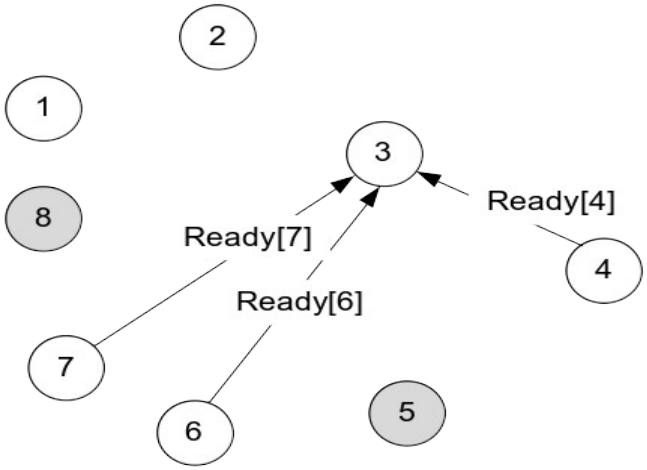
13. **Локальное состояние: L[M, send, receive]** состояние узла (идентификаторы принятых (receive) и отправленных(send) сообщений), вычисляется относительно предыдущего локального состояния, идентифицируется маркером **M[S,V,I]**, вызвавшим его формирование.
14. **Инициатор:** фиксирует свое локальное состояния и оправляет маркер **M[S,V,I]** во все исходящие каналы.
15. **Узел:** получает маркер **M[S,V,I]**, если нет локального состояния с идентификатором S, то формирует его; если есть локальное состояние, то формирует локальное состояние относительно предыдущего с идентификатором **M[S,V,I];** отправляет маркер, во все выходные узлы; отправляет инициатору локальное состояние.
16. **Continue**[**M**] – ответ на маркер **M[S,V,I]**, требующий, повторного запроса локального состояния.
17. **Ready**[**M**] – ответ на маркер **M[S,V,I]**, указывающий на завершение вычислений в узле.
18. **Узел:** получает маркер **M[S,V,I]** от узла **A** , если с предыдущего состояния не поступали новые сообщения от узла **A**, формирует локальное состояние и оправляется маркер **M[S,V,I]** во все выходные узлы; если сообщения от **A** поступали, то возвращает отправившему маркер узлу **A** сообщение **Continue**[**M**].
19. **Узел:** если возвращается **Ready[M]** со всех выходных узлов, в узел приславший маркер оправляется ответ **Ready[M].**
20. **Узел:** если возвращается **Continue**[**M**] с выходного канала, в соответствующий узел снова отправляется маркер **M[S,V,I].**
21. **Инициатор:** если получил **Ready[M]** со всех узлов, то распределенное вычисление завершилось, если из некоторых узлов пришло **Continue**[**M**], то повторная отправка **M[S,V,I].**
22. **Инициатор:** может быть несколько с разными идентификаторами и идентификаторами маркеров.
23. **Распределенная система:** работа не приостанавливается.

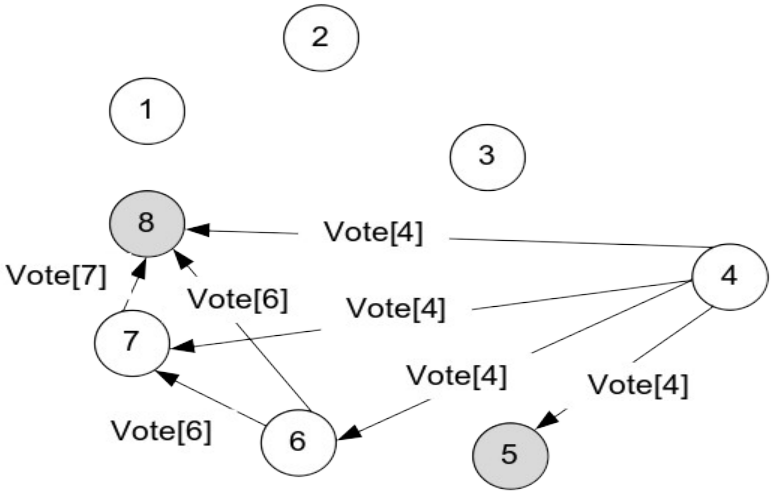
24. **Алгоритм формирования глобального состояния:** реализуется как отдельный протокол системы промежуточного уровня.
25. **Координатор:** компонент (узел) распределенной системы, имеющий специальное назначение (инициатор, арбитр, хранитель централизованной информации). Пример: арбитр BIOS over TCP/IP (хранение таблицы символических имен компьютеров).
26. **Алгоритмы голосования**: алгоритмы выбора арбитра; все узлы одинаковы; каждый имеет идентификатор и знает идентификаторы всех остальных узлов и не знает: какие из этих узлов работают; необходимо определить общий для всех узел, который будет выступать в качестве арбитра.
27. **Алгоритм забияки (bully algorithm)**: выбор процесса с самым большим идентификатором.
28. **Алгоритм забияки:** любой из узлов (инициатор), обнаруживших, что координатор не отвечает, запускает голосование.
29. **Алгоритм забияки:** если узел-инициатор имеет самый большой идентификатор, то он объявляет себя координатором и рассылает всем сообщение **ICoordinator[I]**, где I-собственный идентификатор.
30. **Алгоритм забияки:** узел-инициатор отправляет всем другим узлам с большими, чем у него идентификаторами сообщение **Vote[I]**, где I-собственный идентификатор.
31. **Алгоритм забияки:** если узел получает сообщение **Vote[I]** от узла с меньшим номером, то он отвечает отправителю сообщением **IReady[I]**, где I-собственный идентификатор и отправляет всем узлам с большими, чем у него идентификаторами, сообщение **Vote[I].**
32. **Алгоритм забияки:** если узел-инициатор, не получает ни одного ответа **IReady[I]** на отправленное **Vote[I]**, то он объявляет себя координатором и рассылает всем сообщение **ICoordinator[I].**
33. **Алгоритм забияки:** вновь подключившийся узел, ищет координатора, опрашивая все узлы сообщением **GetCoordinator[I]**, где I-собственный идентификатор и

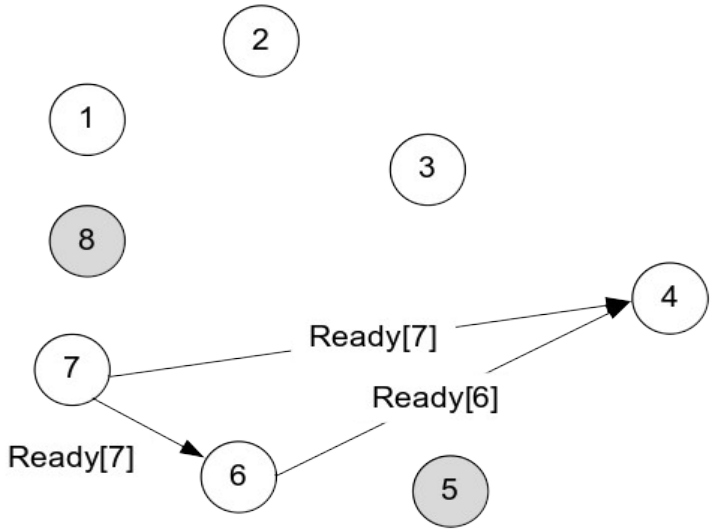
ждет ответа **ICoordinator[I];** если координатор не отвечает, то запускается процесс голосования.

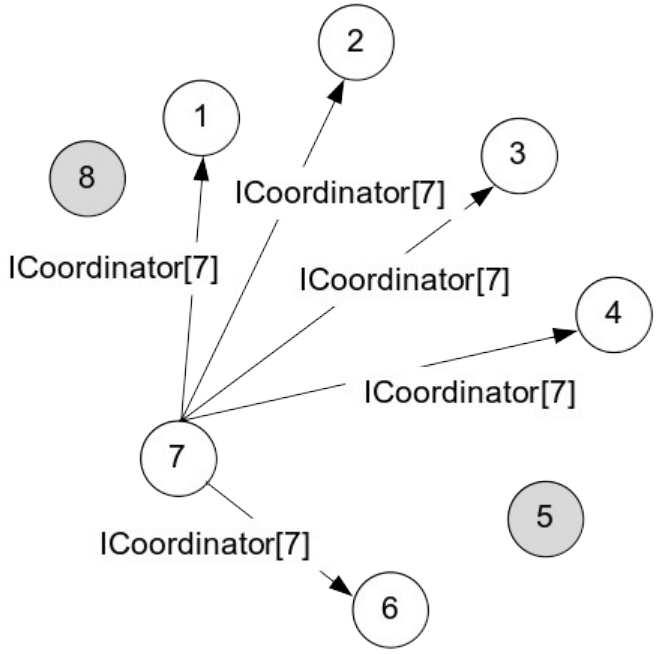
34. **Алгоритм забияки:** пример








2
1
3
8
4
Ready[7]
7
Ready[6]
Ready[7]
6
5


2
1
3
8
ICoordinator[7]
ICoordinator[7]
ICoordinator[7]
4
ICoordinator[7]
7
5
ICoordinator[7]
6