

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

Государственное автономное образовательное учреждение

Высшего образования города Москвы

«Московский городской педагогический университет»

(ГАОУ ВО МГПУ)

Институт цифрового образования

Лабораторная работа №4

«Обнаружение отказов в распределенной системе»

Выполнила: Татаринова Екатерина Михайловна

Группа: АДЭУ-221

Преподаватель: доцент Босенко Т.М.

Москва 2024

## Вариант 15. Оптимизация для географически распределенных систем

**Задача:** найти оптимальные настройки для географического распределенной системы с учетом межгрупповых задержек

gossip interval	gossip fanout	nodes	packet loss	node failures	время до «Хотя бы один узел знает»	время до «Все живые узлы знают»	макс. Использование полосы пропускания, бит/с
0,2	3	200	5%	5%		58.00	26,615,808.00
0,2	3	200	10%	5%		58.00	25,214,976.00
0,2	5	200	5%	5%		34.60	44,359,680.00
0,2	5	200	10%	5%		34.60	42,024,960.00
0,5	3	200	5%	5%		145.00	10,615,323.20
0,5	3	200	10%	5%		145.00	10,085,990.40
0,5	5	200	5%	5%		86.50	17,743,872.00
0,5	5	200	10%	5%		86.50	16,809,984.00
1,0	3	200	5%	5%		290.00	5,323,161.60
1,0	3	200	10%	5%		290.00	5,042,995.20
1,0	5	200	5%	5%		173.00	8,871,936.00
1,0	5	200	10%	5%		173.00	8,404,992.00

### Анализ полученных результатов:

#### Gossip Interval

- Более короткий интервал (0.2 с) увеличивает частоту обмена сообщениями, что может снизить задержку передачи данных, но с другой стороны увеличивает нагрузку на сеть, особенно при высоких потерях пакетов.
- Более длинный интервал (1.0 с) снижает нагрузку, но может замедлить распространение информации.

## Gossip Fanout

- Большее значение fanout (5) позволяет узлам общаться с большим количеством соседей, что может быть полезно для быстрого распространения информации, но также увеличивает сетевой трафик.
- Меньшее значение fanout (3) создает меньшее сетевое взаимодействие, но может замедлить распространение информации.

## **Оптимальные настройки для географического распределенной системы с учетом межгрупповых задержек**

### 1. Gossip Interval:

Значение 0.2 обеспечивает наименьшее время до "Все живые узлы знают". Более высокие значения 0.5 и 1.0 приводят к значительному увеличению времени.

### 2. Gossip Fanout:

Установка Gossip Fanout = 5 также дает наилучшие результаты по времени (34.60 с 0.2 интервалом). В то время как для Fanout = 3 время значительно больше (58.00)

### 3. Потеря пакетов:

Потеря пакетов при 5% гарантирует лучшее время и максимальное использование полосы пропускания по сравнению с 10%. Например, при Gossip Interval = 0.2 и Fanout = 5, значение максимального использования полосы пропускания составляет 44,359,680 бит/с при 5% потерь, в то время как при 10% это значение снижается до 42,024,960 бит/с.

Тогда можно сделать вывод, что оптимальными настройками для минимизации времени до "Все живые узлы знают" и максимизации использования полосы пропускания являются:

- **Gossip Interval: 0.2**
- **Gossip Fanout: 5**
- **Nodes: 200**
- **Packet Loss: 5%**
- **Node Failures: 5%**
- Время до "Все живые узлы знают": 34.60
- Максимальное использование полосы пропускания: 44,359,680 бит/с

Эти настройки обеспечивают наилучшее сочетание скорости и эффективности, что является критически важным для географически распределенных систем, находящихся под влиянием межгрупповых задержек.

№5

**а) Как изменение Gossip Interval влияет на время конвергенции и использование полосы пропускания?**

Уменьшение Gossip Interval, то есть сокращение времени между сообщениями о состоянии между узлами, может привести к более быстрому обмену информацией и, соответственно, к более быстрому времени конвергенции. Однако слишком частые обновления могут также увеличивать нагрузку на сеть и использование полосы пропускания, что может привести к задержкам и, в конечном итоге, к замедлению работы системы. Наоборот, увеличение интервала может сократить использование полосы пропускания, но при этом замедлит время, необходимое для достижения согласия между узлами.

**б) Какое влияние оказывает увеличение Gossip Fanout на производительность системы?**

Увеличение Gossip Fanout, то есть количества узлов, с которыми каждый узел делится информацией, может привести к более быстрым обновлениям состояния системы, так как информация распространяется быстрее. Однако это также может увеличить нагрузку на сеть, особенно если количество узлов велико, что может вызвать накладные расходы и повлиять на производительность системы.

**в) Как масштабируется система при увеличении количества узлов?**

При добавлении узлов система часто может сохранять или даже повышать свою производительность благодаря децентрализованной архитектуре. Однако, если количество узлов слишком велико, может возникнуть проблема с управлением состоянием, что иногда приводит к увеличению времени конвергенции и платы за коммуникацию между узлами. Эффективность масштабирования зависит от алгоритма консенсуса и особенностей реализации.

**№7. Сделайте выводы о влиянии каждого параметра на производительность системы и предложите оптимальные настройки для различных сценариев использования**

**Выводы о влиянии каждого параметра на производительность системы:**

**1. Gossip Interval:**

- Влияние: Более короткий интервал (например, 0.2) приводит к более быстрому распространению информации между узлами, что уменьшает время, необходимое для того, чтобы все узлы узнали данные. Однако это также может увеличить нагрузку на сеть.

- Рекомендация: Для сценариев, где важно быстрое обновление информации (например, в системах мониторинга), следует использовать меньший интервал.

## 2. Gossip Fanout:

- Влияние: Увеличение числа узлов, с которыми обмениваются данные (fanout), улучшает скорость распространения информации и снижает время до "Все живые узлы знают". Но это может также привести к более высокому расходу пропускной способности.

- Рекомендация: Для высоконагруженных систем (например, в реальных приложениях, где нужно передавать данные между многими узлами) лучше использовать более высокий fanout (например, 5).

## 3. Nodes:

- Влияние: Увеличение числа узлов при фиксированных условиях может как уменьшить, так и увеличить время распространения, в зависимости от других параметров (особенно интервала и fanout).

- Рекомендация: Для систем с меньшим числом узлов можно использовать более высокие значения gossip interval и lower fanout. Для масштабируемых систем (большее количество узлов) следует оптимизировать параметрами, чтобы сохранить производительность.

## 4. Packet Loss:

- Влияние: Высокая потеря пакетов (например, 10%) значительно ухудшает время синхронизации узлов и общее использование полосы пропускания. Пожарные настройки (например, 5%) показывают наилучшие результаты.

- Рекомендация: Однако, во многих случаях, если ожидаемая потеря пакетов высока, необходимо учитывать это в планировании. В таких сценариях лучше использовать параметры, минимизирующие их влияние.

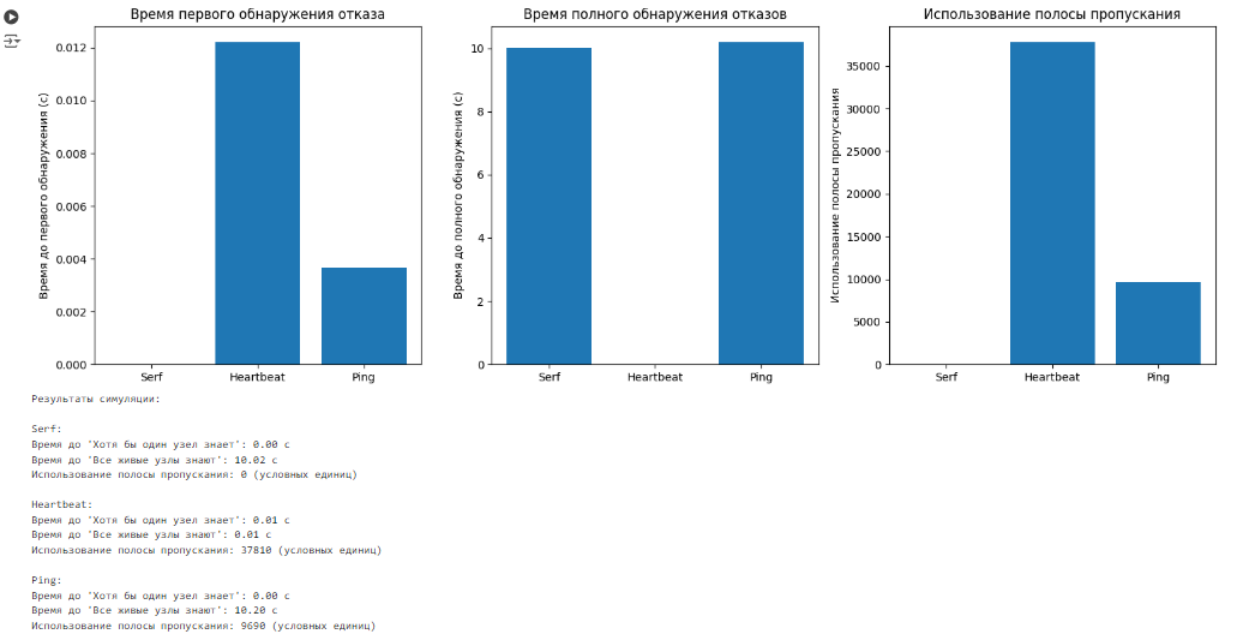
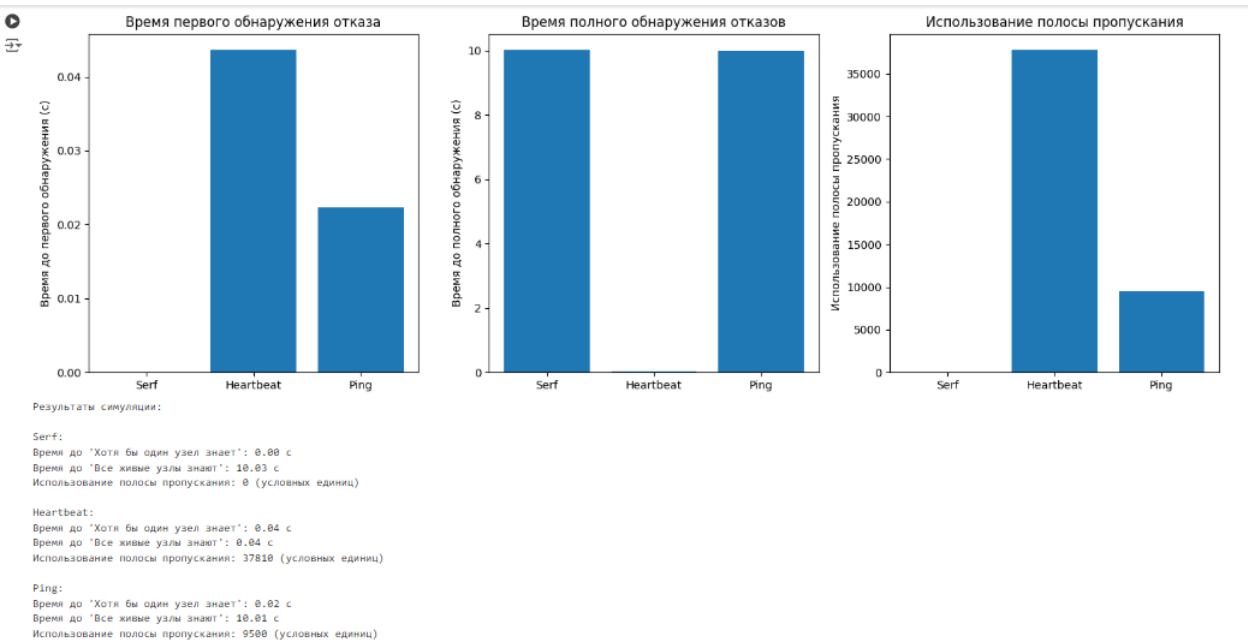
## 5. Node Failures:

- Влияние: Параметр отказов узлов не был непосредственно выражен в данных, но очевидно, что чем выше ожидаемые отказы, тем больший акцент приходится делать на устойчивость системы.
- Рекомендация: Для критически важных систем, работающих в ненадежных сетях, необходимо учитывать возможные отказы узлов и увеличивать размеры fanout и частоту информационных обновлений.

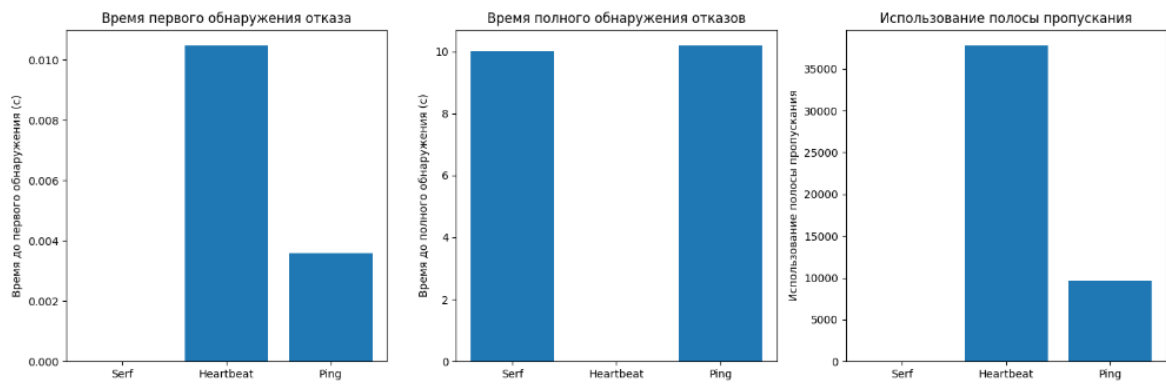
Оптимальные настройки для различных сценариев:

1. Система реального времени (чистая производительность):  
Gossip Interval: 0.2  
Gossip Fanout: 5  
Nodes: 200  
Packet Loss: 5%  
Node Failures: 5%
2. Большая распределённая система (целостность данных):  
Gossip Interval: 0.5  
Gossip Fanout: 3  
Nodes: 500+  
Packet Loss: 5%  
Node Failures: 10% (если узлы могут уходить на длительное время)
3. Система с высокой потерей пакетов (низкая скорость связи):  
Gossip Interval: 1.0  
Gossip Fanout: 3  
Nodes: 200  
Packet Loss: 10%  
Node Failures: 5%

№8. Сравните производительность Serf с другими протоколами обнаружения отказов, такими как heartbeat или ping-based методами.







Результаты симуляции:

Serf:

Время до "Хотя бы один узел знает": 0.00 с  
Время до "Все живые узлы знают": 10.02 с  
Использование полосы пропускания: 0 (условных единиц)

Heartbeat:

Время до "Хотя бы один узел знает": 0.01 с  
Время до "Все живые узлы знают": 0.01 с  
Использование полосы пропускания: 37810 (условных единиц)

Ping:

Время до "Хотя бы один узел знает": 0.00 с  
Время до "Все живые узлы знают": 10.20 с  
Использование полосы пропускания: 9690 (условных единиц)