



РОССИЙСКИЕ ИНТЕРНЕТ-ТЕХНОЛОГИИ  
**Высокие нагрузки**

# Система мониторинга Яндекс

Максим Лапань <[lapan@yandex-team.ru](mailto:lapan@yandex-team.ru)>

Яндекс



# Мониторинг: функции

**• Чем больше серверов вы обслуживаете, тем чаще возникают проблемы.**

**• Присматривать за парой серверов возможно. За десятком неудобно. За сотней уже нереально.**

- Задачи мониторинга:**
- Сбор данных о работе серверов, сервисов, свичей и т.д.;**
- Оповещение о возникновении проблем;**
- Анализ собранных данных (графики).**



# Требования к мониторингу

- **Прозрачная масштабируемость до десятков датацентров;**
- **Мониторинг нескольких тысяч устройств в каждом ДЦ;**
- **Обработка нескольких тысяч уведомлений в день.**



## Суровая действительность: nagios

- На серверах по cron запускаются скрипты с проверками, результаты отправляются на сервера nagios;
- Конфигурация nagios — большой-большой файл, генерируемый скриптами с описанием серверов, сервисов, групп, правил оповещения и т.д.;
- Помимо nagios, данные хранятся в rrd, по которым строятся графики.





# **Nagios: недостатки**

- Система не отказоустойчива и масштабируется переносом части проверок на отдельные сервера;
- Все изменения конфигурации выполняются правкой файлов конфигурации с последующим перезапуском nagios (~ 10-15 минут);
- Слишком большой интервал между проверками и замерами параметров;
- RRD усредняет данные, поэтому невозможно сказать, каково было точное значение параметров, например, месяц назад.
- Каждую из этих проблем в отдельности можно было бы решить. Однако решив все, получим почти полностью переписанный nagios.



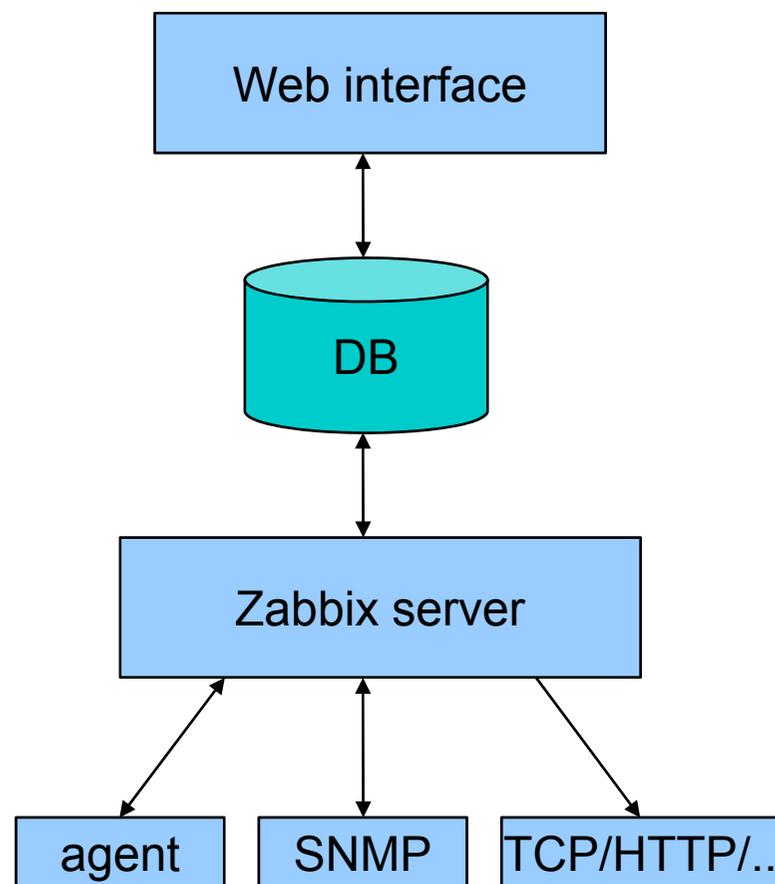
## **Светлое будущее: zabbix**

- **Особенности:**
- **Вся конфигурация хранится в базе, управляется через web-интерфейс;**
- **Единая точка доступа для пользователей;**
- **Разграничение доступа к данным и конфигурации;**
- **Минимальный интервал между замерами — 1 секунда;**
- **С серверов собираются не результаты проверок (сломалось или нет), а количественные характеристики работы, которые анализируются на стороне сервера;**
- **Время хранения данных ограничено лишь дисковым пространством;**
- **Развитые возможности анализа собранных данных.**



## Архитектура zabbix

- **Zabbix agent** – многопоточный демон, собирающий требуемые параметры на машине и отправляющий результаты на сервер;
- **Zabbix server** – собирает данные, выполняет проверки и рассылает уведомления
- **Web interface** – изменение параметров мониторинга, визуализация данных, управление оповещениями, и т.д.





## **Недостатки**

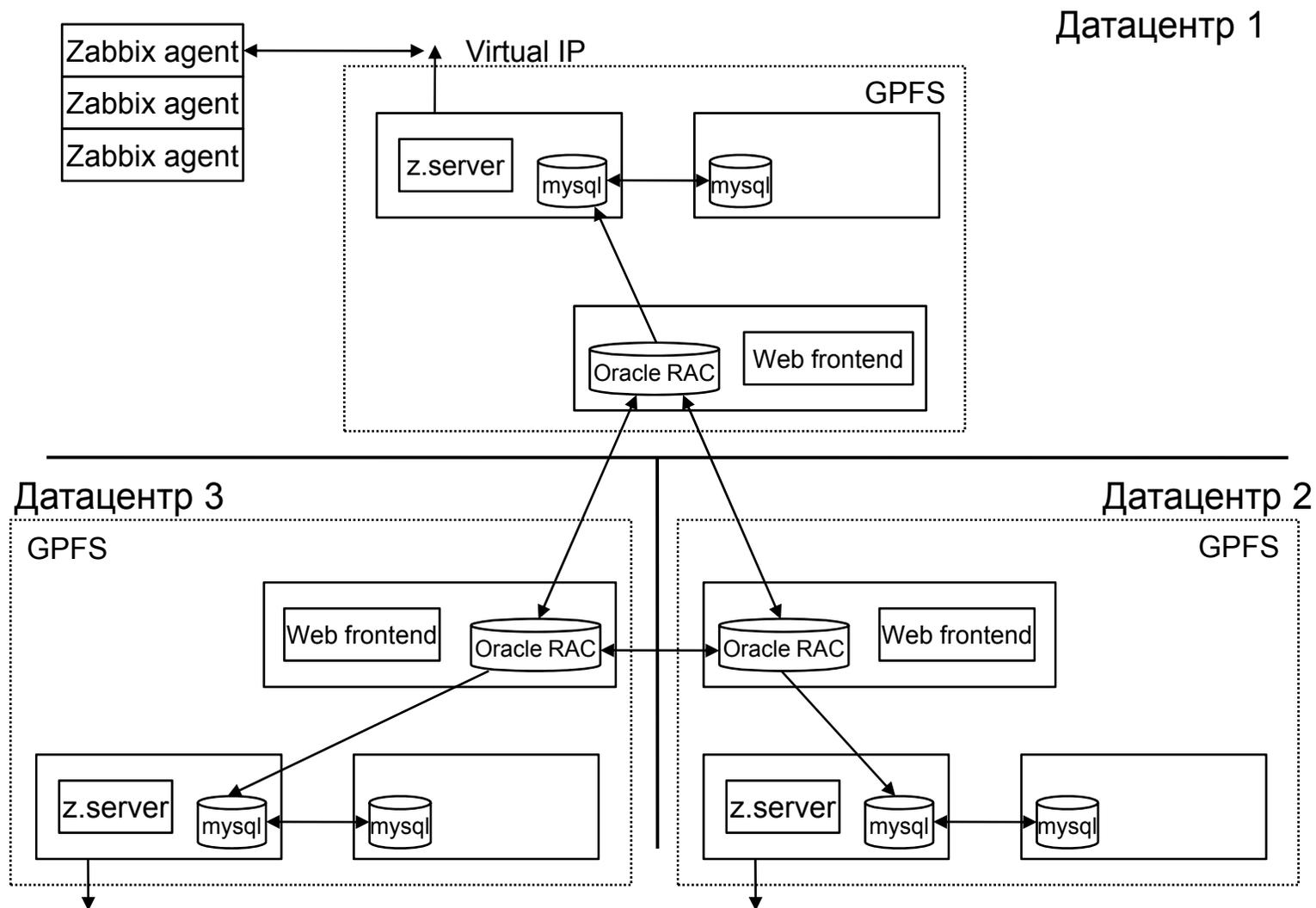
- **Все данные истории хранятся в базе, что неэффективно и ограничивает масштабируемость;**
- **Не обеспечивается отказоустойчивость.**

### **Что делать?**

- **Хранить историю в распределенной FS;**
- **Сооружать отказоустойчивую конфигурацию.**



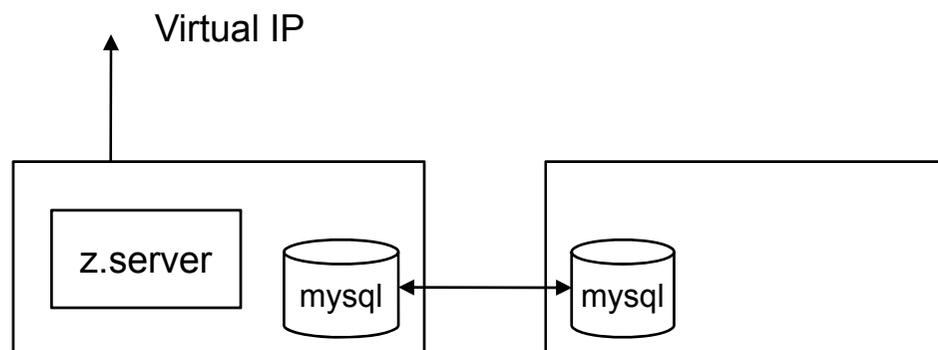
# Яндекс-zabbix





# Мониторинговая пара

- Два сервера, mysql master-master репликация;
- Переключение virtual IP выполняется средствами uCARP;
- В среднем переключение происходит за 1-2 секунды, прозрачно для агентов;
- На активной машине пары используется локальный memcached;
- Каждый сервер пары полностью автономен, то есть может выполнять мониторинг датацентра и отправку уведомлений даже в случае потери связности между ДЦ.

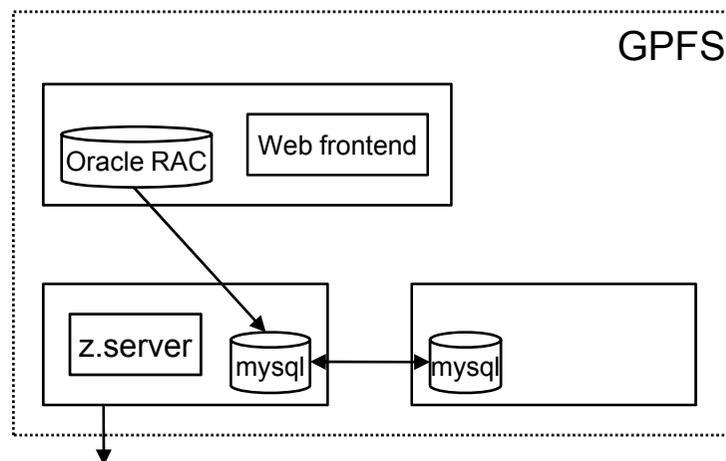




# Хранение истории

**Для хранения данных и состояния мониторинговой пары используется GPFS. В кластер GPFS входят обе машины пары и сервер с web-интерфейсом.**

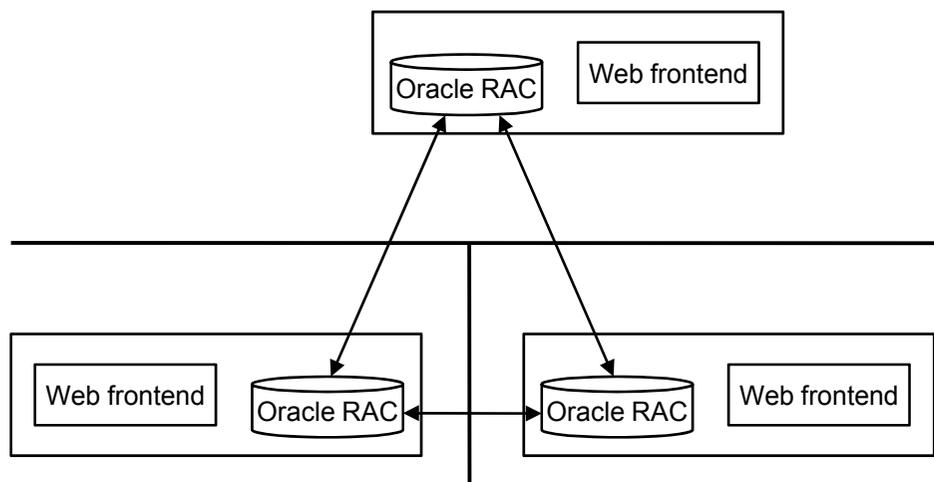
**Хранятся все данные всегда. За год в результате детального мониторинга 1000 серверов (30 параметров с сервера, интервал обновления 5 секунд) накапливается 1.5 Тб данных.**





# Web-интерфейс

**Web-морда балансируется по ДЦ и использует локальный кусочек RAC. При модификации данных в базе конфигурации срабатывают специальные триггера, выполняющие асинхронную репликацию изменений в mysql на мониторинговых парах. Таким образом, в каждом ДЦ хранится отдельная реплика конфигурации мониторинга.**





# Переделки в zabbix

- **Хранение исторических данных в файловой системе. В базе хранится только конфигурация;**
- **Механизм асинхронной репликации конфигурации с oracle в mysql во все датацентры;**
- **Разделение ответственности zabbix-серверов по датацентрам;**
- **Поддержка в zabbix-сервер virtual IP;**
- **Улучшения в интерфейсе по работе с большим количеством серверов и проверок.**



РОССИЙСКИЕ ИНТЕРНЕТ-ТЕХНОЛОГИИ  
**Высокие нагрузки**

# Final

## **Спасибо за внимание!**

**Максим Лапань <[lapan@yandex-team.ru](mailto:lapan@yandex-team.ru)>**