The background features a complex pattern of glowing blue light trails that swirl and curve across a black field, creating a sense of dynamic movement and energy.

**Разработка
высоконагруженных проектов,
Что нового?**

Олег Бунин



HighLoad ++

Конференция разработчиков
высоконагруженных систем

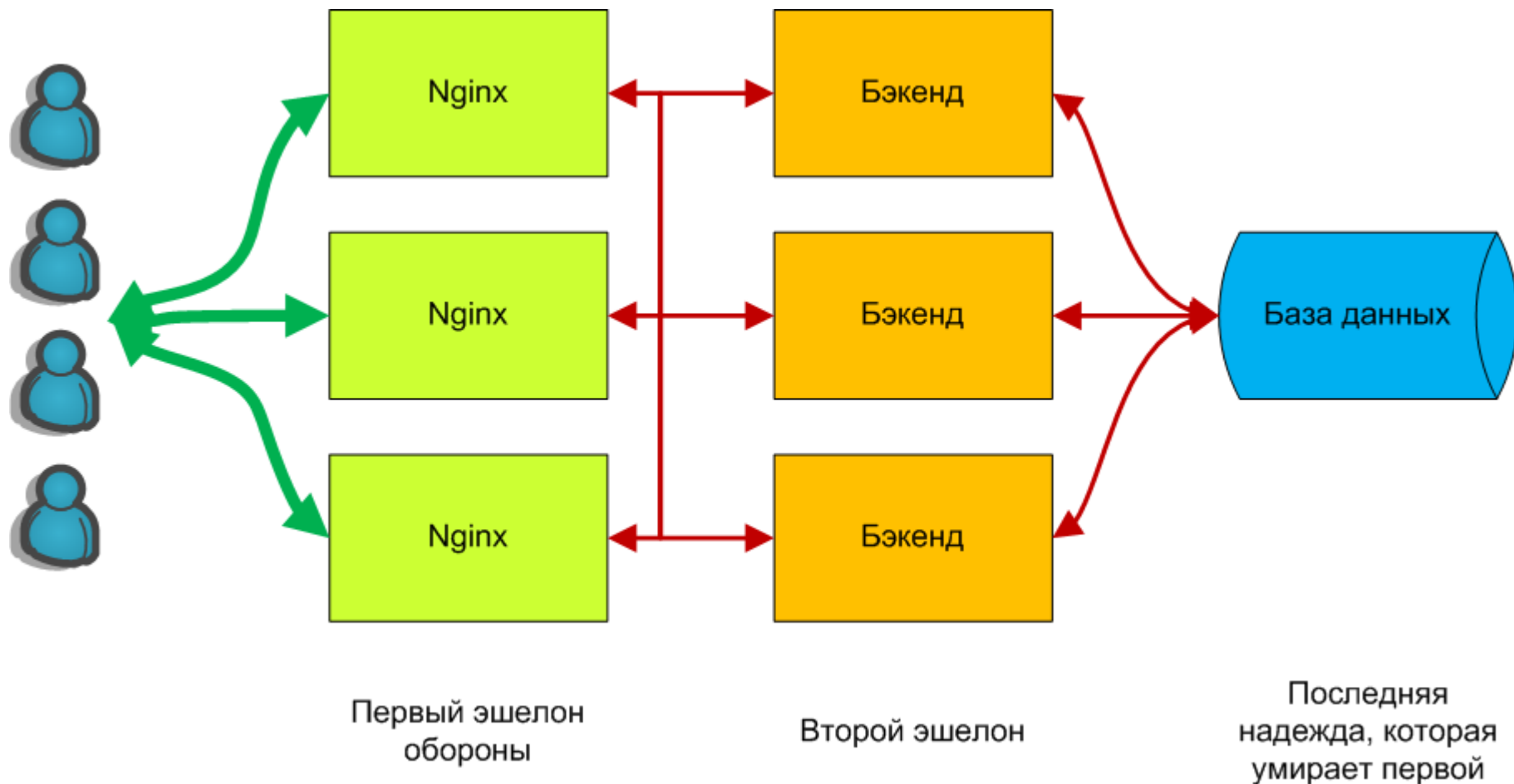
- Масштабируемые и отказоустойчивые архитектуры;
- Алгоритмы работы, системы массового обслуживания, системы реального времени;
- Быстрые веб-сервера;
- Тестирование, функциональное и нагрузочное;
- Защита от DDOS атак;
- Оптимизация баз данных, кластерные файловые системы;
- Поддержка, мониторинг, деплоймент;
- Тюнинг операционных систем.

24 и 25 октября, Москва, 700 участников

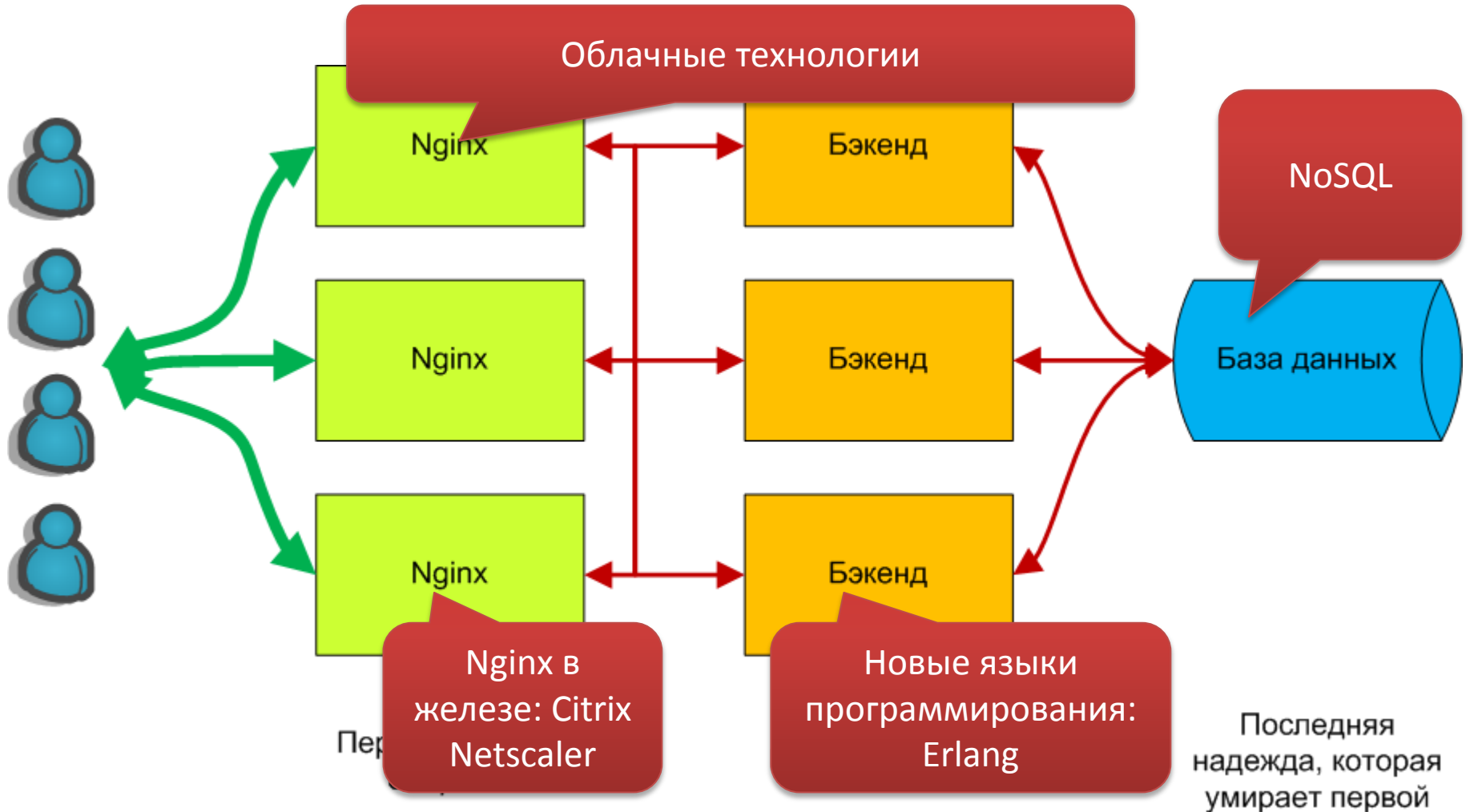
<http://www.highload.ru/>

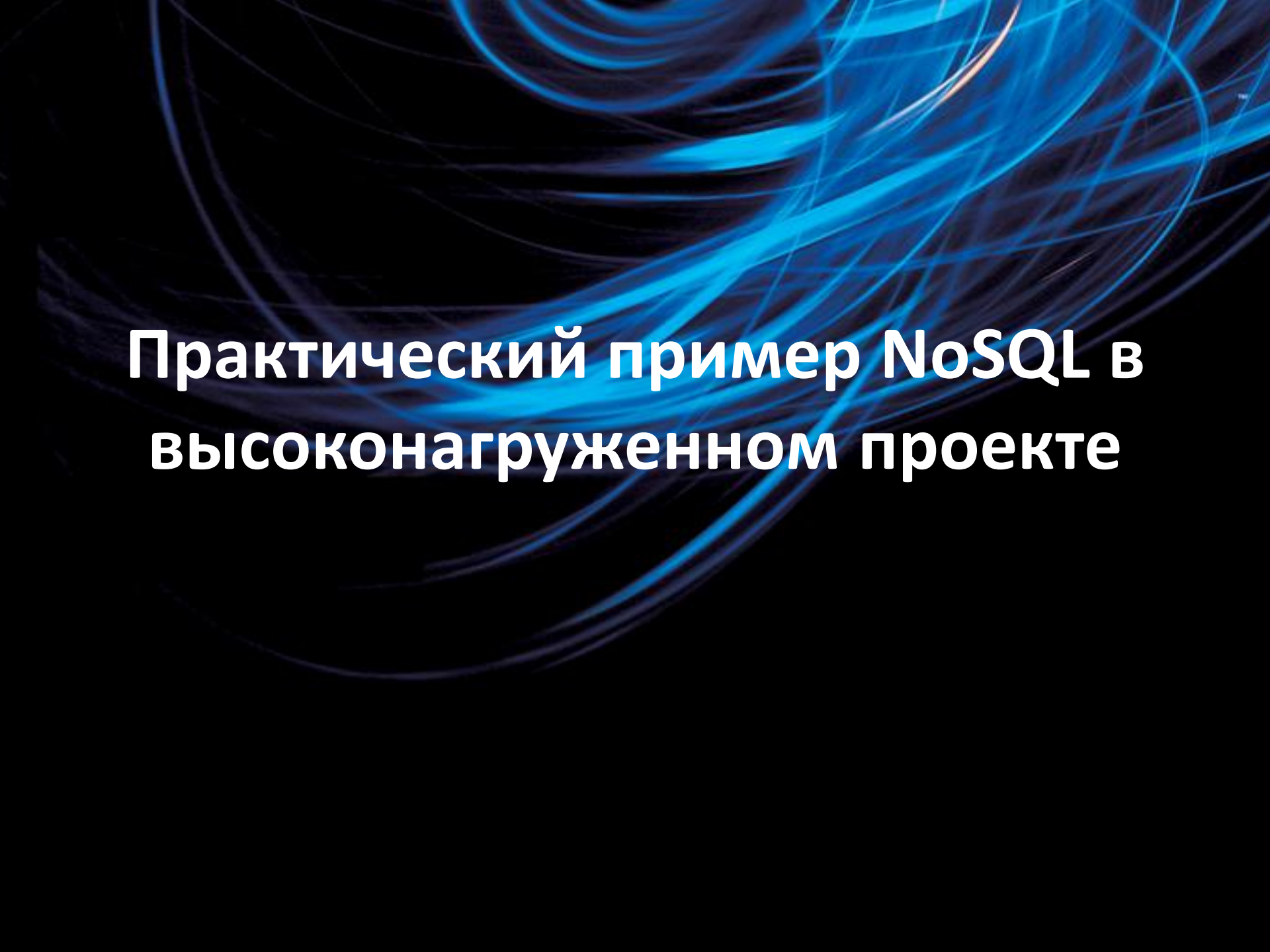


Трехзвенная структура



Трехзвенная структура



The background features a dynamic pattern of glowing blue light trails and streaks, creating a sense of motion and energy against a solid black background. The trails are most concentrated in the upper right and middle sections, with some fainter lines extending towards the bottom left.

Практический пример NoSQL в высоконагруженном проекте

Модели данных

- **Реляционная модель:** данные в базе данных представляют собой набор отношений;
- **Иерархическая модель:** база данных состоит из объектов с указанием отношений родитель ↔ ребенок;
- **Сетевая модель:** база данных со структурой в виде графа;
- **Объектно-ориентированная модель:** база данных, в которой данные представлены в виде моделей объектов.

Разработка чата

Messages

+ New Message



Search Messages



Nate Gadgibalaev

Слушай, я хотел сделать лично, но я теряю надежду на встречу! Была пара моме...

2 seconds ago | ● ×



Ирина Куриленко

Заходите на мой сайт -мобилингус - очень нужны посетители, сайт посвящен мо...

March 28 | ○ ×



Olga Lebedeva

ну да. 😊 ну как бы я сама отчасти сдурила и занялась благотворительностью т...

March 22 | ○ ×



Kevin Weil Friends with Leif Hedstrom and Robert Johnson

← Kevin, hello! Did you recieve my emails with invitation to speak at conference "Rus...

March 21 | ○ ×



Leif Hedstrom

Oh man, it's all in my spam folder 😞. Let me take a look at it tonight, and I'll email ...

March 21 | ○ ×



Elena Masolova

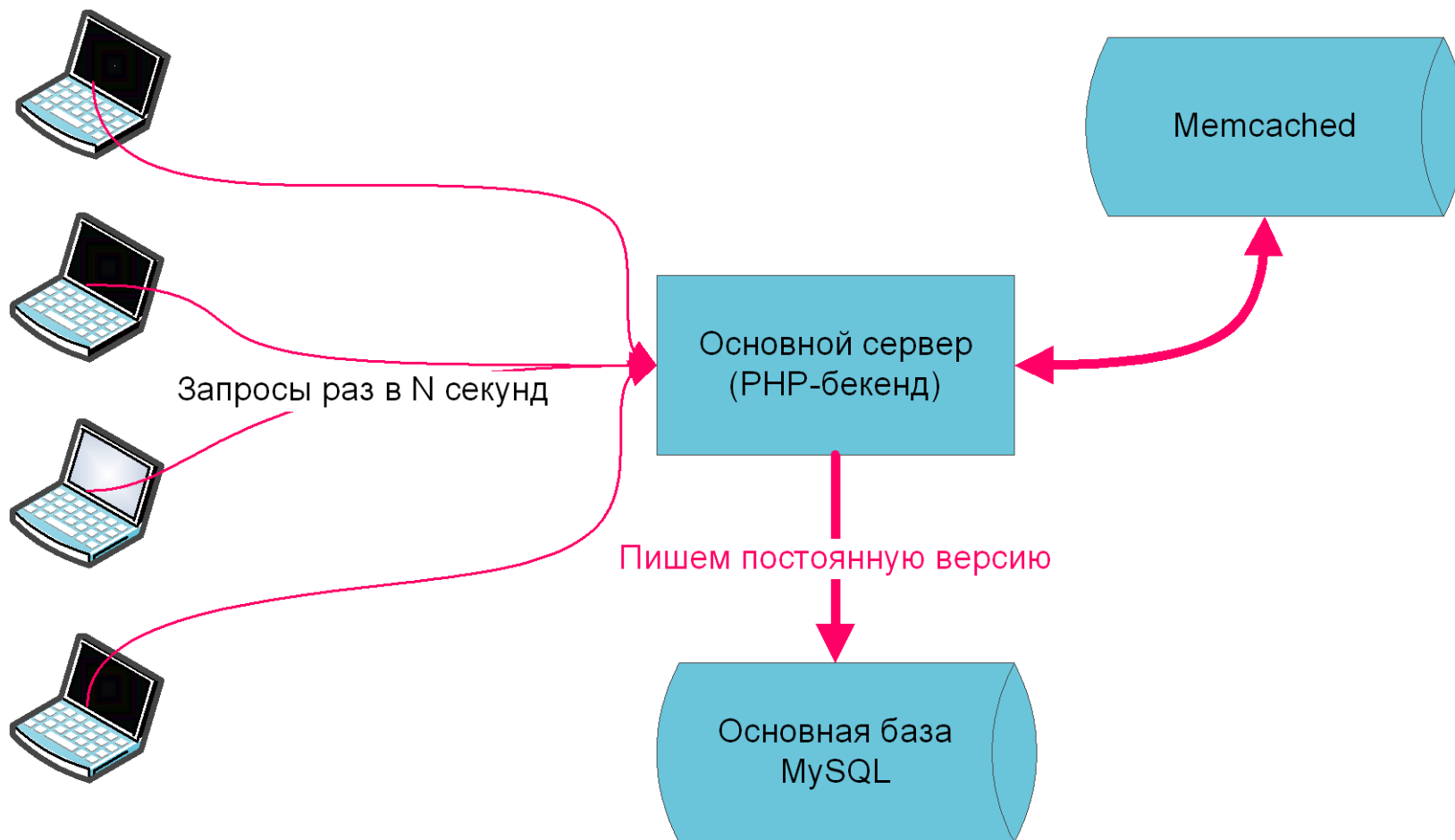
← Елена, спасибо, позвал СТО Зинги 😊

March 17 | ○ ×

Постановка задачи

- Около десяти тысяч одновременных подключений;
- Доставка сообщения в реальном времени (чат);
- Сохранение истории сообщений (переписка);
- Одновременная переписка с несколькими участниками.

Обычная схема



Вопросы к обычной схеме

Кэширование
переписок

- Устаревание кэша
- Чистка кэша

Обновление
информации

- Конкурентные обновления

**Мгновенная
доставка**

- Только эмуляция – опрос сервера каждую секунду



Мгновенность доставки

Как нам сообщать клиенту о том, что пришло сообщение?

Нам нужен event-driven!



Event-driven

Нам нужно сообщать в реальном времени клиенту о новых сообщениях! Используем для этого событийно-ориентированную модель.

Мгновенную доставку можно решить некоей имитацией длинных незакрытых соединений (WebSockets, FlashSocket, AJAX Long polling, Iframe).

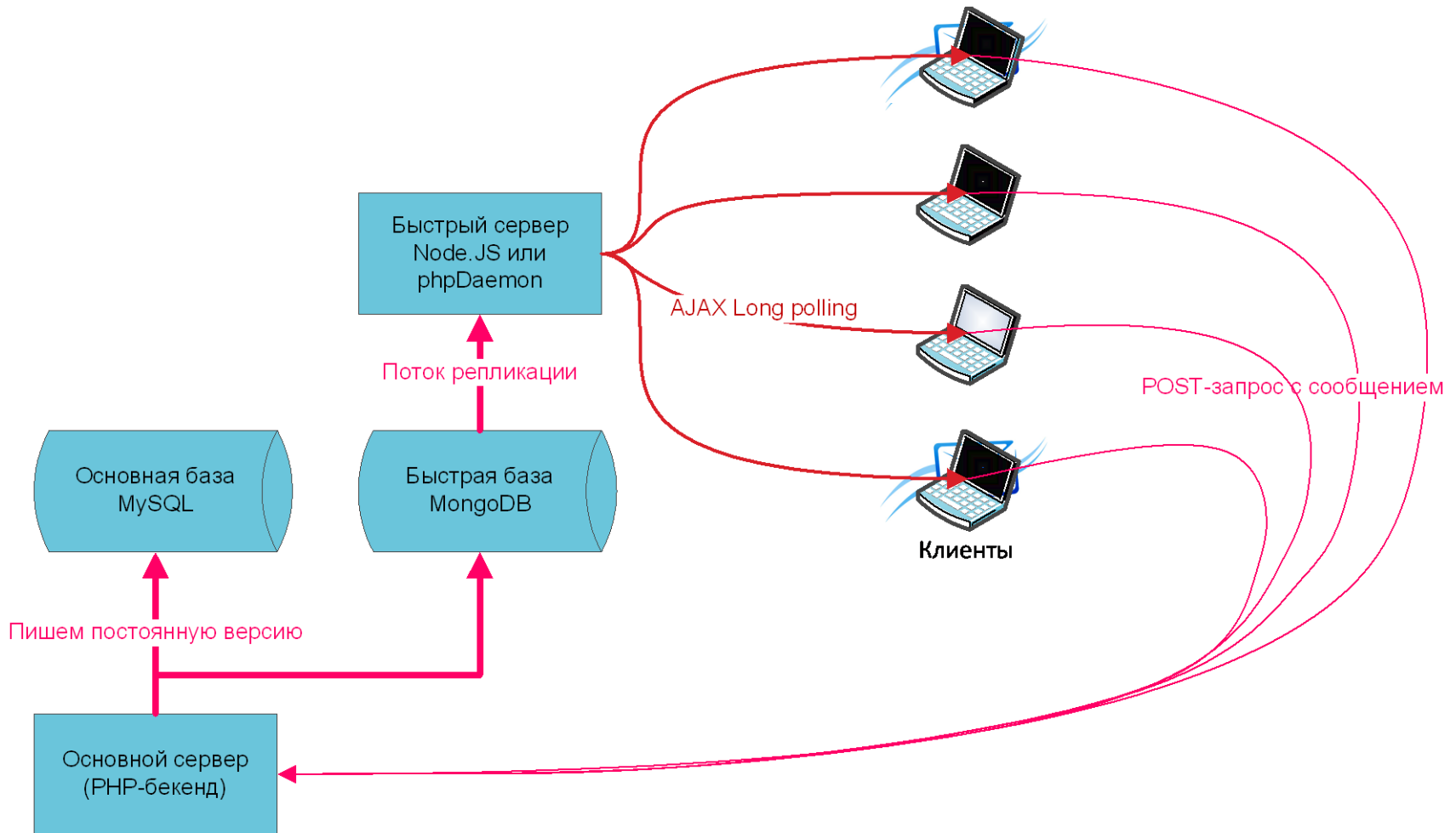
Реклама!

Если Вы поняли, что я сейчас сказал –
приходите ко мне на работу в новый
питерский офис!

Присылай резюме на **oleg.bunin@ontico.ru**

Прямо сейчас! Сделай это!

Event-driven чат



Event-driven чат

- Каждый новый клиент, входящий в переписку устанавливает соединение с одним из «быстрых» серверов (**phpDaemon** или **Node.js**) и от него же будет получать сообщения.
- При отправке на основной сервер нового сообщения, оно записывается в постоянную базу данных (**MySQL**), а так же в некую «быструю» базу данных (в данном случае **MongoDB**), где будет храниться только актуальная переписка, например за последние сутки.
- Master MongoDB записывает изменения в коллекцию **local.oplog.\$main** для репликации.
- «Быстрые сервера» в это время ведут опрос «быстрой» базы на предмет наличия в ней новых сообщений, и, если такие есть, отправляют их тому клиенту, которому они адресованы.

Тестирование скоростей

Демон	Complete requests	Requests per second	Time per request, ms (across all concurrent requests)	Min request, ms	Max Request, ms
<i>10 keep-alive соединений</i>					
phpDaemon	1147	54.36	18.440	21	7116
Node.JS	1285	64	15.618	16	9064
<i>100 keep-alive соединений</i>					
phpDaemon	1543	75.15	13.313	22	12889
Node.JS	1284	64.12	15.765	17	13347
<i>500 keep-alive соединений</i>					
phpDaemon	1365	67.73	14.824	15	15174
Node.JS	1236	61.71	16.611	23	16078
<i>1000 keep-alive соединений</i>					
phpDaemon	1159	56.99	17.680	19	13103
Node.JS	1528	75.02	13.354	17	16785

Красным цветом отмечены «лучшие показатели»



oleg.bunin@ontico.ru