

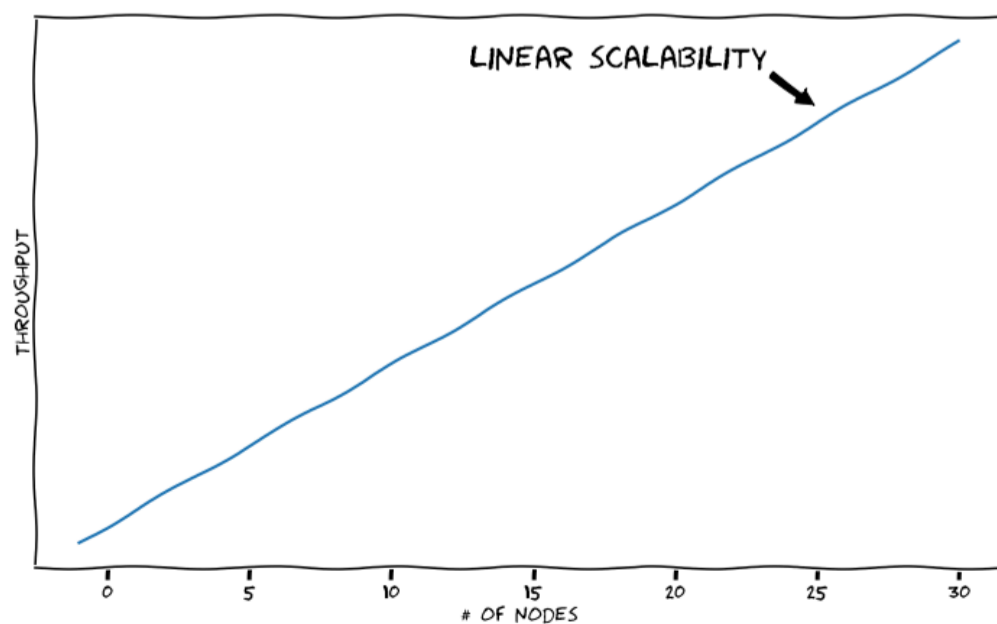
Текстовая расшифровка видео:

ЧУТЬ БОЛЬШЕ О РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

План:

- Линейное масштабирование;
- Contention component;
- Crosstalk component;
- Проблема византийских генералов;
- Консенсус и консистентность.

Линейное масштабирование



$$Throughput(N) = \frac{\lambda N}{1} \quad \lambda - \text{coefficient of performance}$$

Одни из частых вопросов при выборе БД или моделировании распределенной системы, чтобы

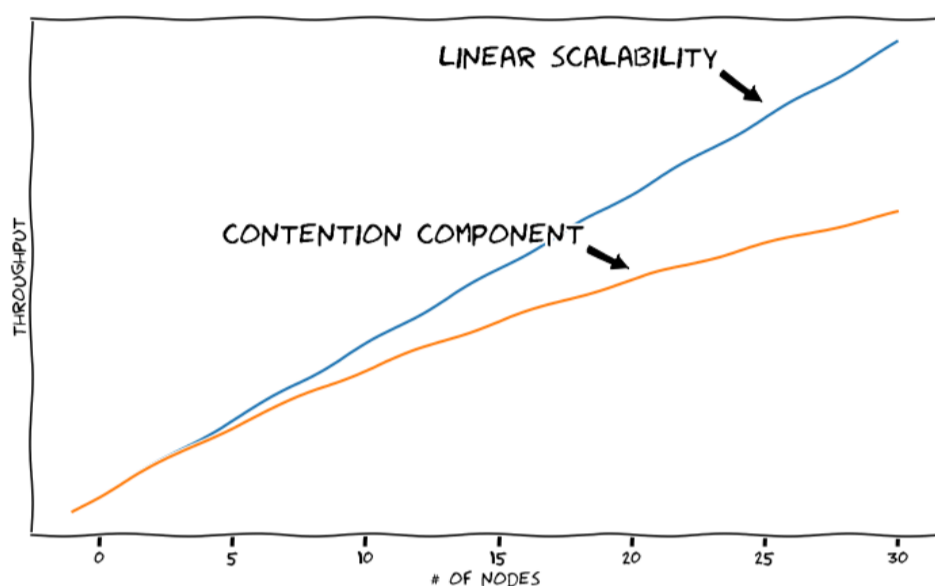


процессить Big data, следующие:

- Как все будет масштабироваться?
- Как выбрать количество и мощность серверов, чтобы все адекватно работало?

Один из базовых выводов следующий: если какая-то система заявляет вам, что линейно масштабируется, то вам нагло врут. Чтобы это работало, все машины должны быть абсолютно независимы друг от друга, существовать в полном вакууме и никогда не обмениваться данными как между собой, так и с центральной нодой. Такую систему в реальном мире представить очень сложно, так как практически всегда возникает Contention component.

Contention component

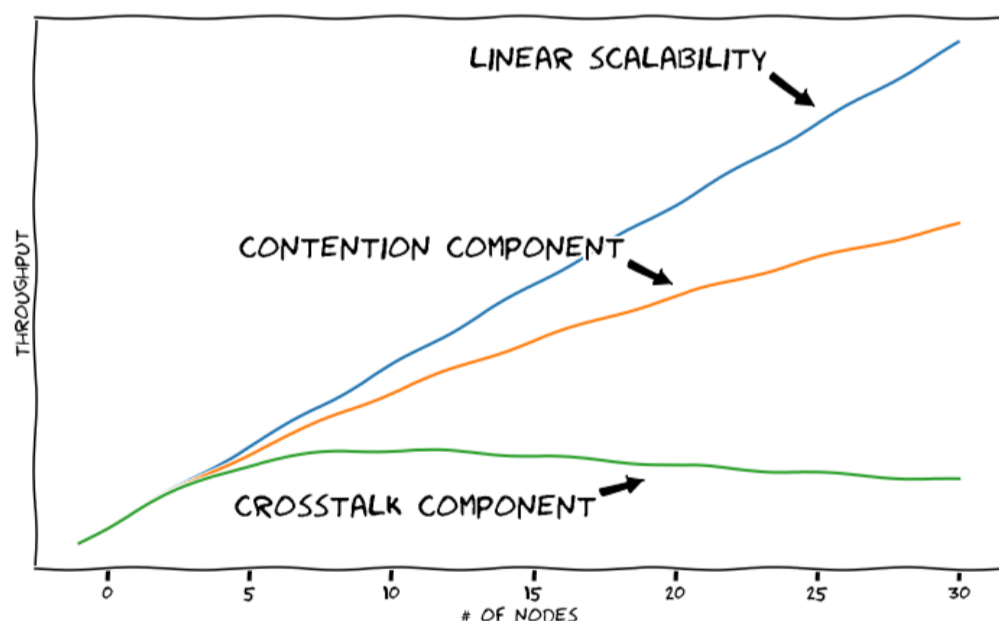


$$Throughput(N) = \frac{\lambda N}{1 + \sigma(N-1)}$$

σ – coefficient of serialization (contention)

Если это какая-то распределенная база, вы делаете к ней запросы через отдельную машину. Соответственно, все данные собираются на фолловерах и обсчитываются, после чего результат пересылается на машину, из которой вы сделали запрос. Всегда есть что-то, во что упирается производительность системы.

Crosstalk component



$$Throughput(N) = \frac{\lambda N}{1 + \sigma(N-1) + kN(N-1)}$$

k – crosstalk coefficient

На практике все будет еще хуже. Дело в том, что в вашей системе будут шкафы, будет горизонтальное взаимодействие между нодами. В реальных распределенных системах/базах данных, может возникнуть ситуация, при которой добавление новых машин не только не ускорит систему, но и замедлит ее.

Данная формула имеет название «**Универсальный закон масштабирования**» (Universal scalability law (USL)). Эта часть базовой теории распределенных систем, которая относится к теории массового обслуживания (в англ. версии – «Теория очередей»).

Проблема византийских генералов

Мы не раз говорили о важности обеспечения консистентности данных и т.д. Данная проблема широко известна в контексте распределенных систем и имеет название «**Проблема византийских генералов**».

Проблема византийских генералов формулируется следующим образом:

Есть генералы, которые стоят на холмах. Холмы окружены неприятелем. Для того, чтобы атаковать, им нужно скооперироваться и договориться, в какое время им наносить удар. Однако для того, чтобы обмениваться какой-либо информацией, необходимо отправить гонца с холма на холм через поле боя. С большей долей вероятности гонца убьют.

Возникает вопрос: могут ли генералы прийти к четкому консистентному состоянию?

Естественно, это метафора о пересылке пакетов через ненадежную сеть, где машины могут умирать, а сетевая связность рваться и т.д.

Математически ответом на возникший вопрос будет «Нет». Достижение полного консенсуса между всеми «генералами» невозможно. Однако мы можем решить сложившуюся проблему в достаточно хорошем приближении.

Консенсус и консистентность

Существует ряд алгоритмов, который это организует:

- [Paxos](#) (старше, сложнее);
- [Raft](#) (новее, проще);
- [CRDT](#).

Данные алгоритмы уже реализованы. Их используют множество баз и распределенных систем. Они часто вынесены в отдельные компоненты. Так, например, мы уже говорили о ZooKeeper, который реализует алгоритм консенсуса для проектов в экосистеме Hadoop. В мире Kubernetes существует etcd (реализует рафт и обеспечивает консенсус между разными компонентами). Также существует CRDT (это про то, как договориться о том, что хранится в определенной структуре данных между большим количеством сетевых агентов).

В современных системах, когда речь идет о больших масштабах, может использоваться blockchain.

Blockchain – это механизм обеспечения достоверной валидации.

Как вам урок?



Изучил, далее >

Слёрм ©

+7 (495) 248-05-80

[Лицензия №ДЛ-1368 от 22.08.2019](#)

[Политика конфиденциальности](#)

[Публичная оферта](#)

