

1

Что такое доступность

Никого не интересуют замечательные возможности вашей системы, если ее невозможно использовать.

Одним из важнейших аспектов архитектуры распределенных систем является доступность. Хотя есть такие компании и сервисы, для которых временная недоступность оправданна и допустима, большая часть фирм не может позволить себе ни секунды недоступности, так как ее последствия неизбежно повлияют на удовлетворенность клиентов и, как следствие, на финансовое благополучие фирмы.

Далее приведены фундаментальные вопросы, которыми должны задаваться все компании, которые хотят определить важность доступности систем для себя и своих клиентов. Эти вопросы и очевидные на них ответы лежат в основе того, почему доступность критична для крупномасштабных приложений.

- ❑ Зачем кому-то покупать ваш сервис, если он не будет работать в нужный момент?
- ❑ Что думают или чувствуют ваши клиенты, когда у них возникает потребность в нашем сервисе, а он не работает?
- ❑ Как удовлетворить наших клиентов, заработать средства для компании, выполнить обязательства и требования, когда сервис недоступен?

Клиенты будут довольны вашим продуктом и заинтересованы в нем только тогда, когда система работает. Существует прямая и весьма

существенная корреляция между доступностью системы и удовлетворенностью клиентов.

Высокая доступность настолько важна для построения масштабируемых систем, что я посвящаю этой теме значительную часть книги. Как создать систему (сервис, приложение, среду) с высокой доступностью, когда к ней предъявляется и без того широкий спектр требований?

В этой главе мы определим, что такое доступность и как она соотносится с надежностью. Это будет использоваться в последующих главах, когда мы начнем обсуждать, какую роль доступность играет в проектировании высокомасштабируемых приложений.

Большая игра

Сегодня воскресенье — день важного матча. Вы пригласили 20 близких друзей на просмотр матча на новом 300-дюймовом телевизоре. Все пришли, дом полон еды и пива. Все веселятся. Игра вот-вот начнется. И...

...пропадает электричество;

...телевизор гаснет;

...для вас и ваших друзей матч окончен.

В раздражении вы снимаете трубку и набираете номер энергетической компании. Представитель компании, не проявляя ни капли сочувствия, сообщает: «Приносим извинения, но мы гарантируем только 95%-ную доступность электросети».

Почему важна доступность? Потому, что клиенты рассчитывают на то, что ваши сервисы работают постоянно. Доступность менее 100 % может обернуться катастрофой для вашего бизнеса.

Доступность и надежность

Доступность и надежность — похожие, но очень разные понятия. Очень важно понимать разницу между ними.

В нашем контексте надежность в общем случае означает качество системы. Обычно так называют способность системы стабильно

работать в соответствии со спецификациями. Программное обеспечение считается надежным, если оно проходит набор тестов и делает то, что от него ожидается.

В нашем контексте доступность в общем случае означает способность системы выполнять задачи, которые она способна выполнять. Есть ли система? Работает ли она? Отвечает ли она на запросы? Утвердительный ответ на все три вопроса означает, что система доступна.

Как видим, доступность и надежность очень похожи. Систему трудно назвать доступной, если она ненадежна. Систему трудно назвать надежной, если она недоступна.

Однако, когда речь идет о надежности программного обеспечения, обычно говорят о способности программного обеспечения исполнять возложенные на него обязанности. В общем, основным индикатором доступности является то, проходит ли программное обеспечение все наборы тестов.

Кроме того, когда мы имеем в виду доступность системы, мы говорим о том, что система развернута и работоспособна. Ответит ли она на запрос, который я отправил?

Вот что мы подразумеваем при использовании этих терминов.

- ❑ *Надежность.* Способность системы выполнять требуемые действия, не допуская ошибок.
- ❑ *Доступность.* Готовность системы к работе при необходимости выполнить требуемые действия.



Система, которая при сложении 2 и 3 получает 6, характеризуется плохой надежностью. Система, которая при сложении 2 и 3 не возвращает ответа, характеризуется плохой доступностью. Надежность может быть улучшена путем тестирования. Доступность улучшить обычно гораздо сложнее.

Можно привести в программу ошибку, которая приведет к тому, что $2 + 3$ будет равно 6. Такую ошибку легко отловить и исправить с помощью набора тестов.

Допустим, есть приложение, которое надежно выдает результат $2 + 3 = 5$. А теперь представим, что оно запущено на компью-

тере с нестабильным сетевым подключением. Каков результат? Когда вы запускаете приложение, оно иногда возвращает 5, а иногда не возвращает ничего. Приложение может быть надежным, но при этом недоступным.

Эта книга в основном посвящена созданию систем с высоким уровнем доступности. Предположим, что ваша система надежна. Допустим также, что вы знаете, как создавать и запускать наборы тестов. Мы будем обсуждать надежность только тогда, когда она непосредственно влияет на архитектуру системы или ее доступность.

Что ухудшает доступность

Что заставляет приложение, которое раньше работало нормально, демонстрировать низкую доступность? Причин много.

- ❑ *Исчерпание ресурсов.* При увеличении количества пользователей или объема данных, требуемых системе, ваше приложение может стать жертвой исчерпания ресурсов, что приведет к снижению производительности и замедлению работы.
- ❑ *Незапланированные изменения, связанные с увеличением нагрузки.* Рост популярности приложения может потребовать внести изменения в код, чтобы обеспечить поддержку более высокой нагрузки. Эти изменения часто реализуются на скорую руку, в последний момент, без должного планирования и продумывания, что увеличивает вероятность появления ошибок.
- ❑ *Увеличение количества подвижных частей.* По мере того как приложение набирает популярность, часто необходимо подключать к его поддержке все больше и больше разработчиков, проектировщиков, тестировщиков и других людей. Множество работающих над приложением создают большое количество подвижных частей в результате добавления новых или изменения существующих возможностей или обычного технического обслуживания. Чем больше людей трудится над приложением, тем больше в нем подвижных частей и тем больше вероятность их некорректного взаимодействия.

- ❑ *Внешние зависимости.* Чем сильнее ваше приложение зависит от внешних ресурсов, тем больше оно подвержено проблемам, связанным с их доступностью.
- ❑ *Технический долг.* Усложнение приложений обычно ведет к росту технического долга (накоплению желаемых изменений в программном обеспечении и ожидающих исправления багов, связанному с ростом и развитием приложения). Технический долг увеличивает вероятность появления проблем.

Все быстрорастущие приложения сталкиваются с одной, несколькими или всеми перечисленными проблемами. Вследствие этого в приложениях, ранее работавших безупречно, могут появиться проблемы с доступностью. Часто они подкрадываются незаметно и возникают неожиданно.

Но большинство растущих приложений имеют одну и ту же проблему: они со временем начинают испытывать сложности с доступностью.

Проблемы с доступностью стоят денег вам и вашим клиентам. Они стоят доверия и преданности клиентов. Ваша компания долго не просуществует, если у нее будут постоянные проблемы с доступностью.

Создание масштабируемых приложений означает создание приложений с высоким уровнем доступности.