Предисловие

В облачном сообществе и за его пределами eBPF стал одной из самых горячих технических тем за последние годы. Новое поколение мощных инструментов и проектов в области сетевой работы, безопасности, наблюдения и многого другого было создано (и продолжает создаваться) с использованием eBPF в качестве платформы, предлагая более высокую производительность и точность по сравнению с их предшественниками. Конференции, связанные с eBPF, такие как eBPF Summit и Cloud Native eBPF Day, привлекли тысячи участников и зрителей, и на момент написания этой статьи сообщество eBPF Slack насчитывает более 14 000 участников.

Почему eBPF выбран в качестве базовой технологии для столь многих инфраструктурных инструментов? Как это обеспечивает обещанные улучшения производительности? Чем полезен eBPF в таких разрозненных технических областях, которые варьируются от отслеживания производительности до шифрования сетевого трафика?

Эта книга призвана ответить на эти вопросы, дав читателю представление о том, как работает eBPF, а также предоставив введение в написание кода eBPF.

Для кого эта книга

Эта книга предназначена для разработчиков, системных администраторов, операторов и студентов, интересующихся eBPF и желающих узнать больше о том, как он работает. Он послужит основой для тех, кто хочет самостоятельно изучить написание программ eBPF. Поскольку eBPF предоставляет прекрасную платформу для совершенно нового поколения приборов и инструментов, разработчики eBPF, вероятно, найдут оплачиваемую работу в ближайшие несколько лет.

Но вам не обязательно планировать самостоятельное написание кода eBPF, чтобы эта книга была вам полезна. Если вы работаете в сфере эксплуатации, безопасности или в любой другой роли, связанной с программной инфраструктурой, вы, вероятно, столкнетесь с инструментами на основе eBPF сейчас или в ближайшие несколько лет. Если вы что-то понимаете во внутреннем устройстве этих инструментов, у вас будет больше возможностей для их эффективного использования. Например, если вы знаете, как события могут запускать программы eBPF, у вас будет лучшая ментальная модель того, что именно измеряет инструмент на основе eBPF, когда он показывает вам показатели производительности. Если вы разработчик приложений, вы также можете столкнуться с некоторыми из этих инструментов на основе eBPF — например, если вы настраиваете производительность приложения, вы можете использовать такой инструмент, как Parca, для создания графиков пламени, показывающих, какие функции выполняются. самое время. Если вы оцениваете инструменты безопасности, эта книга поможет вам понять, в чем проявляется преимущество eBPF и как избежать его наивного использования, которое менее эффективно против атак.

Даже если вы сегодня не используете инструменты eBPF, я надеюсь, что эта книга даст вам интересные сведения о тех областях Linux, о которых вы, возможно, раньше не задумывались. Большинство разработчиков воспринимают ядро как должное, так как они используют языки программирования с удобными высокоуровневыми абстракциями, которые позволяют им сосредоточиться на разработке приложений, что достаточно сложно! Они используют такие инструменты, как отладчики и анализаторы производительности, чтобы эффективно выполнять свою работу. Знание того, как работает отладчик или средство

повышения производительности, может быть интересным, но не обязательным. Тем не менее, для многих из нас забавно и приятно спуститься в кроличью нору, чтобы узнать больше[1]. Точно так же большинство людей будут использовать инструменты eBPF, не беспокоясь о том, как они устроены. Артур Кларк писал, что «любая достаточно продвинутая технология неотличима от магии», но лично мне нравится копаться и выяснять, как работает магический трюк. Вы можете быть похожи на меня и чувствовать себя обязанным изучить программирование eBPF, чтобы лучше понять, что возможно с этой технологией. Если да, я думаю, вам понравится эта книга.

Что охватывает эта книга

eBPF продолжает развиваться довольно быстрыми темпами, что затрудняет написание исчерпывающего справочника, который не нуждается в постоянном обновлении. Однако есть некоторые основы и базовые принципы, которые вряд ли претерпят существенные изменения, и именно об этом и пойдет речь в этой книге.

Глава 1 устанавливает сцену, описывая, почему eBPF является такой мощной технологией, и объясняя, как возможность запуска пользовательских программ в ядре операционной системы обеспечивает множество замечательных возможностей.

Все станет более конкретным в главе 2, где вы увидите несколько примеров «Hello World», которые познакомят вас с концепциями программ и карт eBPF.

В главе 3 более подробно рассматриваются программы eBPF и то, как они работают в ядре, а в главе 4 исследуется интерфейс между приложениями пользовательского пространства и программами eBPF.

Одной из больших проблем eBPF в последние годы был вопрос совместимости между версиями ядра. В главе 5 рассматривается подход «компиляция один раз, запуск везде» (CO-RE), который решает эту проблему.

Процесс проверки — пожалуй, самая важная характеристика, отличающая eBPF от модулей ядра. Я познакомлю вас с верификатором eBPF в главе 6.

В главе 7 вы познакомитесь со многими различными типами программ eBPF и точками их подключения. Многие из этих точек подключения находятся в сетевом стеке, и в главе 8 более подробно рассматривается применение eBPF для сетевых функций. В главе 9 рассматривается, как eBPF используется для обеспечения безопасности. инструменты.

Если вы хотите написать приложение пользовательского пространства, взаимодействующее с программами eBPF, вам может помочь множество библиотек и фреймворков. В главе 10 дается обзор опций для различных языков программирования.

Наконец, в главе 11 я загляну в свой хрустальный шар и расскажу вам о некоторых будущих разработках, которые, вероятно, произойдут в мире eBPF.

Необходимые знания

В этой книге предполагается, что вы знакомы с основными командами оболочки в Linux и с идеей использования компилятора для преобразования исходного кода в исполняемую

1 На парижской конференции dotGo в 2017 году я выступила с докладом, в котором показала, как работает отладчик (https://www.youtube.com/watch?v=TBrv17QyUE0).

программу. Есть несколько простых примеров извлечений из файлов Makefile, при условии, что вы имеете хотя бы минимальное представление о том, как make использует эти файлы.

Есть много примеров кода на Python, C и Go. Вам не нужно глубокое знание этих языков, чтобы извлечь что-то из этих примеров, но вы извлечете максимум пользы из книги, если в целом с удовольствием почитаете какой-нибудь код. Я также предполагаю, что вы знакомы с идеей указателей, которые идентифицируют в памяти расположение.

Пример кода и упражнения

В этой книге много примеров кода. Если вы хотите попробовать их сами, вы найдете соответствующий репозиторий GitHub и инструкции по установке и запуску кода по адресу https://github.com/lizrice/learning-ebpf.

Я также включил упражнения в конце большинства глав, чтобы помочь вам изучить программирование eBPF путем расширения примеров или написания собственных программ.

Поскольку eBPF постоянно развивается, доступные вам функции зависят от используемой версии ядра. Многие ограничения, применимые к более ранним версиям, были подняты или ослаблены в более поздних версиях. В проекте Iovisor есть полезный обзор версий ядра, в которые были добавлены различные функции BPF, и в этой книге я попытался отметить, когда были добавлены конкретные возможности, которые я описываю. Примеры были протестированы с использованием ядра версии 5.15, и на момент написания этой статьи некоторые популярные дистрибутивы Linux еще не поддерживали такую последнюю версию ядра. Если вы читаете эту книгу вскоре после ее публикации, вы можете обнаружить, что некоторые функции не будут работать в ядре Linux, используемом в вашей организации в производстве.

eBPF только для Linux?

eBPF изначально разрабатывался для Linux. Нет особых причин, по которым тот же подход нельзя было бы использовать и в других операционных системах — действительно, Microsoft разрабатывает реализацию eBPF для Windows. Я кратко обсуждаю это в главе 11, но в остальной части книги я сосредоточусь только на реализации Linux, и все примеры будут взяты из Linux.

Условные обозначения, используемые в этой книге

В этой книге используются следующие типографские соглашения:

Курсив

> Указывает новые термины, URL-адреса, адреса электронной почты, имена файлов и расширения файлов.

Моноширинный текст

> Используется для листинга программ, а также внутри абзацев для ссылки на элементы программы, такие как имена переменных или функций, базы данных, типы данных, среда переменные, операторы и ключевые слова.

Полужирный моноширинный шрифт

Показывает команды или другой текст, который пользователь должен ввести буквально.

Курсив моноширинный

Показывает текст, который следует заменить значениями, заданными пользователем, или значениями, определяющими добывается по контексту.

Использование примеров кода

Дополнительный материал (примеры кода, упражнения и т. д.) доступен для загрузки по адресу https://github.com/lizrice/learning-ebpf.

Если у вас есть технический вопрос или проблема с использованием примеров кода, отправьте электронное письмо по адресу bookquestions@oreilly.com.

Эта книга предназначена для того, чтобы помочь вам выполнить свою работу. В общем, если к этой книге прилагается пример кода, вы можете использовать его в своих программах и документации. Вам не нужно обращаться к нам за разрешением, если вы не воспроизводите значительную часть кода. Например, написание программы, использующей несколько фрагментов кода из этого книга не требует разрешения. Для продажи или распространения примеров из книг O’Reilly требуется разрешение. Чтобы ответить на вопрос, цитируя эту книгу и код примера, разрешения не требуется. Включение значительного количества примеров кода из этой книги в документацию по вашему продукту требует разрешение.

Мы приветствуем, но обычно не требуем указания авторства. Атрибуция обычно включает название, автора, издателя и ISBN. Например: «Изучение eBPF Лиз Райс (O’Рейли). Copyright 2023 Vertical Shift Ltd., 978-1-098-13512-6». Если вы считаете, что использование вами примеров кода выходит за рамки добросовестного использования или разрешений, данных выше, не стесняйтесь обращаться к нам по адресу permissions@oreilly.com.

Онлайн-обучение O’Reilly

Более 40 лет O’Reilly Media предоставляет технологии и бизнес-тренинги, знания и понимание, которые помогут компании преуспевать.

Наша уникальная сеть экспертов и новаторов делится своими знаниями и опытом с помощью книг, статей и нашей онлайн-платформы обучения. Платформа онлайн-обучения O’Reilly предоставляет вам доступ по требованию к обучающим курсам в реальном времени, схемам углубленного обучения, интерактивным средам кодирования и обширной коллекции текстов и видео из O’Reilly и более 200 других издателей. Для получения дополнительной информации посетите https://oreilly.com.

Как контактировать с нами

Комментарии и вопросы, касающиеся этой книги, направляйте издателю:

O’Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

800-998-9938 (in the United States or Canada)

707-829-0515 (international or local)

707-829-0104 (fax)

У нас есть веб-страница для этой книги, где мы перечисляем опечатки, примеры и любые дополнительные сведения. информация. Вы можете получить доступ к этой странице по адресу https://oreil.ly/learning-eBPF.

Напишите по адресу bookquestions@oreilly.com, чтобы прокомментировать или задать технические вопросы по поводу этой книги.

Для получения новостей и информации о наших книгах и курсах посетите https://oreilly.com.

Найдите нас в LinkedIn: https://linkedin.com/company/oreilly-media.

Следите за нами в Твиттере: https://twitter.com/oreillymedia.

Смотрите нас на YouTube: https://youtube.com/oreillymedia.

Благодарности

Я хотела бы поблагодарить многих людей, внесших огромный вклад в написание этой книги:

• Мои технические рецензенты — Тимо Беккерс, Джесс Мэйлс, Квентин Монне, Кевин Шелдрейк и Селеста Стингер — предоставили подробные, действенные отзывы и отличные идеи по улучшению примеров, за что я им очень благодарен.

• Я стою на плечах гигантов, которые создали, популяризировали и продолжают поддерживать eBPF, включая Даниэля Боркманна, Томаса Графа, Брендана Грегга, Андрея Накрыйко, Алексея Старовойтова и многих других, которые внесли свой вклад не только в код, но и в конференцию, переговоры и сообщения в блогах для сообщества.

• Спасибо моим талантливым и милым коллегам из Isovalent, многие из которых являются специалистами по eBPF и ядру, у которых я продолжаю многому учиться.

• Спасибо также команде O’Reilly, особенно моему редактору Рите Фернандо, которая оказала мне бесконечную поддержку в процессе написания книги, а также за планирование, которое помогло уложиться в график; и Джон Девинс, который в первую очередь вдохновил меня на написание книги.

• Фил Перл не только дал полезный отзыв о содержании, но и следил за тем, чтобы я ела и делала перерывы. Я всегда благодарен ему за поддержку и ободрение.

Я также хочу поблагодарить всех замечательных людей, которые на протяжении многих лет находили время, чтобы оставлять ободряющие комментарии о моей работе, будь то лично на мероприятии или в социальных сетях. Невероятно вдохновляет знать, что то, что я написала или записала, помогло кому-то еще разобраться с технической концепцией или дало их желание построить или написать что-то самостоятельно. Спасибо!