



O que é
observabilidade?





A observabilidade ajuda os desenvolvedores a entender arquiteturas de várias camadas: o que está lento, o que está quebrado e o que precisa ser feito para melhorar o desempenho.

Ao tornar os sistemas observáveis, qualquer pessoa da equipe pode navegar facilmente do efeito à causa em um sistema de produção. O caminho do efeito à causa geralmente requer muitas etapas, incluindo qualquer número de intermediários inocentes. A observabilidade é um meio de seguir cada uma dessas etapas.

Nas palavras de Shaun McCormick, engenheiro sênior da BigCommerce, *“observabilidade não é apenas saber que um problema está acontecendo, mas saber por que está acontecendo. E saber como posso entrar e consertar isso.”*

Mais formalmente, a observabilidade é definida como a capacidade de medir o estado interno de um sistema apenas por suas saídas.

Para sistemas distribuídos, como microsserviços, serverless, service mesh, etc., essas saídas são dados de telemetria: logs, métricas e traces.

Dados de telemetria: logs, métricas e rastreamentos

Existem três tipos principais de dados de telemetria através dos quais os sistemas são observáveis.

Logs

Linhas de texto estruturadas ou não estruturadas que são emitidas por um aplicativo em resposta a algum evento no código. Logs são registros distintos de “o que aconteceu” com algum atributo de sistema específico em um momento específico.





Exemplo de log estruturado:

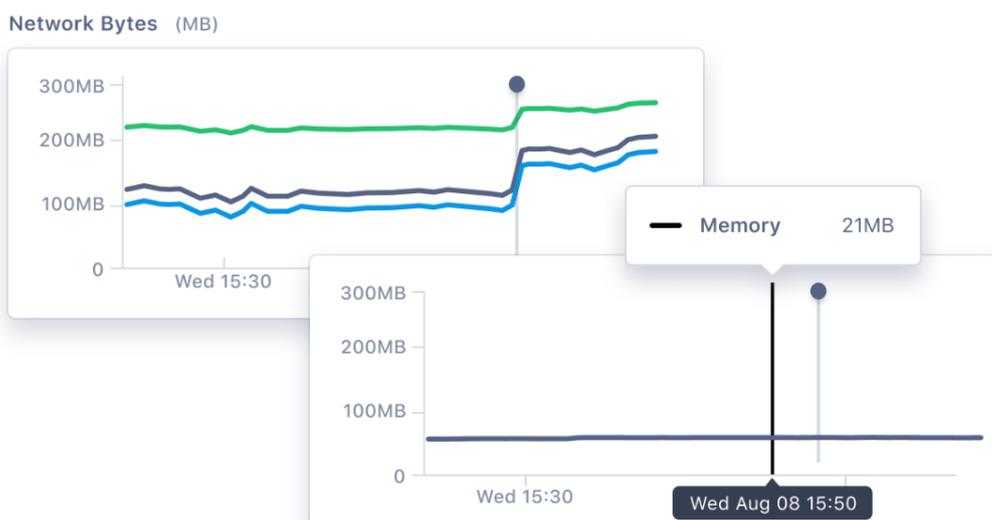
```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

Exemplo de log não estruturado:

```
"thing happened"
```

Métricas

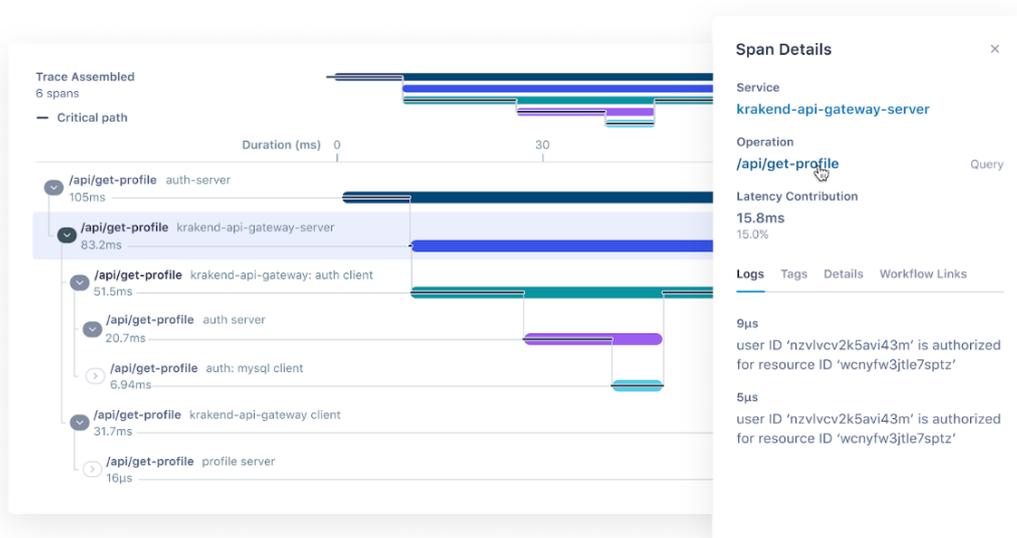
Uma métrica pode informar... POR Uma métrica pode informar o quanto de memória está sendo utilizado por um processo do total ou pelo número de solicitações por segundo que estão sendo tratadas por um serviço.





Traces

Um único trace mostra a atividade de uma transação ou solicitação individual conforme ela flui por um aplicativo. Os traces (ou rastreamentos) são uma parte crítica da observabilidade, pois fornecem contexto para outras telemetrias. Por exemplo, os traces podem ajudar a definir quais métricas seriam mais valiosas em uma determinada situação ou quais logs são relevantes para um determinado problema.



Que perguntas a Observabilidade pode responder?

Esta é, sem dúvida, uma lista incompleta, mas como a aplicação da observabilidade é tão ampla, pensamos que seria útil fornecer uma amostra de possíveis questões que podem ser abordadas por uma solução de observabilidade eficaz:

- Por que x está quebrado?
- De quais serviços a minha aplicação depende — quais serviços dependem da minha aplicação?
- “O que deu errado durante este lançamento?”
- “Por que o desempenho se degradou no último trimestre?”
- “Por que meu pager simplesmente disparou?”





- "O que mudou? Por que?"
- "Quais registros devemos olhar agora?"
- "Devemos reverter este cenário?"
- "Esse problema está afetando alguns usuários do Android ou todos eles?"
- "Como é o desempenho do sistema para nossos clientes mais importantes?"
- Qual SLO devemos definir?
- Estamos fora do SLO?
- Qual era a aparência do meu serviço no momento x?
- Qual era a relação entre meu serviço e x no momento y?
- Qual era a relação de atribuição em todo o sistema antes da implantação? Como é agora?
- O que provavelmente está contribuindo para a latência agora? O que é mais provável que não?
- Essas otimizações de desempenho estão no caminho crítico?

Observabilidade vs. monitoramento

Como a observabilidade é diferente do monitoramento?

O monitoramento exige que você saiba com o **que** se importa antes de saber **quem** se importa com isso. A observabilidade vem da teoria de controle e permite que você entenda todo o seu sistema e como ele se organiza, para que em seguida você utilize essas informações a fim de descobrir com o que especificamente você deve se preocupar, caso algo ocorra.

Por exemplo, quando você está tentando identificar o problema durante um incidente: se houver uma nova dependência profunda na pilha afetando seu serviço, a observabilidade exibirá essas informações. As ferramentas de monitoramento não.

Outra maneira de ver isso: o monitoramento exige que você já saiba o que é normal. Já a observabilidade permite a descoberta de diferentes tipos de "normal" observando como o sistema realmente se comporta, ao longo do tempo, em diferentes circunstâncias.

"Na teoria do controle, observabilidade é uma medida de quão bem os estados internos de um sistema podem ser inferidos pelo conhecimento de suas saídas externas. A





observabilidade e a controlabilidade de um sistema são duas matemáticas.” - Wikipédia

A única constante em sistemas complexos é a mudança, sejam versões de software, tráfego de clientes ou até dependências de terceiros. Esperar que alguém preveja a normalidade nesse ambiente, não é a atitude que prepara a aplicação e a empresa para o sucesso. A observabilidade abraça uma filosofia de consciência e flexibilidade em vez de esperar uma previsão perfeita.

Observabilidade	VS.	Monitoramento
<ul style="list-style-type: none">• Ganha conhecimento ativamente• Realiza perguntas baseadas em hipóteses• Construído para conter ambientes dinâmicos com mudanças complexas• Escolhido pelos desenvolvedores de sistemas com variabilidade e permutações desconhecidas		<ul style="list-style-type: none">• Consome informação de forma passiva• Faz perguntas baseadas em dashboards• Construído para manter ambientes estáticos com poucas variações• Utilizado por desenvolvedores de sistemas com poucas mudanças e permutações conhecidas

Observabilidade vs. APM

Os fornecedores convencionais de monitoramento de desempenho de aplicativos (APM) dependem muito de sampling. Isso significa que, quando você precisa depurar um problema, obtém apenas 1% ou mais dos dados. Claro, o resultado é que há apenas uma chance fracionária de você chegar à causa raiz na primeira tentativa.

Depois que você finalmente encontrou a causa raiz, caso isso exista, há um atraso até que você possa testar novamente. Isso vai arrastar seus esforços de MTTR drasticamente.

Além disso, essas ferramentas usam uma abordagem baseada em agente, que requer CPU. À





medida que você entra no mundo dos serviços, contêineres ou até mesmo vários monólitos, isso exigirá recursos adicionais que você precisará considerar.

Por fim, após “x” meses, seus dados se transformarão em resumos de agregação de dados, o que significa que você não terá mais dados de fidelidade total se precisar auditar suas implantações ou tentar encontrar eficiências em seu código.

Cardinalidade na Observabilidade

Cardinalidade é um termo matemático que nos vem da teoria dos conjuntos, uma teoria fundamental da matemática iniciada na década de 1870 por matemáticos como Georg Cantor. Para simplificar demais, um conjunto é qualquer coleção de elementos únicos, e a cardinalidade desse conjunto é a contagem desses elementos. Mas, o que isso tem a ver com observabilidade?

Por que nos importamos com a cardinalidade?

A cardinalidade dos dados de observabilidade não é um problema que pode ser ignorado: deve ser gerenciado. Existem muitas soluções para esse problema, cujos detalhes não entraremos a fundo aqui, mas o mais importante é entender por que os desafios de cardinalidade são inevitáveis. Uma vez que entendemos isso, então se torna possível discutir como mitigar esses problemas.

1. Em última análise, a cardinalidade é uma consequência inevitável da escala. À medida que os sistemas e aplicativos se tornam cada vez maiores e mais complexos, nossos requisitos para entender esses sistemas se tornam mais sutis e detalhados, não menos.
2. Cardinalidade não é um problema que pode ser resolvido simplesmente jogando mais e mais recursos nele. Geralmente, as combinações exclusivas de métricas de série temporal aumentarão a uma taxa maior do que a quantidade de recursos alocáveis para manter essas métricas na memória para uma consulta. Diferentes componentes do seu sistema também precisarão de soluções diferentes – onde a cardinalidade adicionada pode ser significativamente importante! É





válido considerar serviços compartilhados. Diferentes consumidores do serviço podem querer aplicar seus próprios atributos a medições que outros consumidores podem não precisar. Isso pode adicionar várias complicações à forma como você escreve, implanta, configura e opera seus aplicativos.

3. Se você não estiver preparado para a cardinalidade, poderá facilmente se deparar com situações difíceis para se recuperar. Um único atributo errôneo adicionado a uma métrica amplamente usada pode desencadear repentinamente uma explosão de novas séries temporais sendo geradas e consumidas pelo servidor de métricas. Isso pode causar uma inflação severa no uso de memória não apenas pelos serviços de aplicativos, mas também pelo servidor de métricas e seus componentes associados, pois ele se esforça para acompanhar o aumento da carga. Isso pode ter o infeliz efeito colateral de fazer com que sua infraestrutura de métricas falhe ou se torne extremamente lenta para responder a consultas, deixando você cego em relação aos problemas que estão ocorrendo e removendo uma de suas ferramentas para entender por que o problema está ocorrendo em primeiro lugar.

Que estratégias devemos empregar para abordar a questão crítica da cardinalidade na observabilidade?

1. Pense nas métricas como a borda de sua telemetria. Em geral, o processamento de métricas é rápido – mais rápido que a agregação de logs, mais rápido que a análise de rastreamento agregado. Você quer se apoiar nessa velocidade e garantir que está emitindo métricas para coisas que são os principais indicadores de problemas e você deseja fazer isso de forma consistente em todos os seus serviços.

2. Seja consistente com os atributos. Seus logs, rastreamentos e métricas devem usar as mesmas chaves para o mesmo conceito em vários servidores, sistemas operacionais, estratégias de implantação etc. Isso não apenas facilita o entendimento dos humanos, como reduz a carga cognitiva necessária para entender e interpretar painéis desconhecidos e dados de telemetria emitidos por um serviço dependente.





3. Trate seu código de observabilidade da mesma forma que qualquer outra alteração em seu sistema. Você não mesclaria cegamente uma solicitação pull que alterasse um recurso crítico do seu aplicativo, assim como não deve mesclar cegamente uma solicitação pull que adiciona ou altera métricas existentes. Ao implantar alterações com impacto potencial na cardinalidade da métrica, certifique-se de pegar o pager e garantir que tudo ocorra sem problemas.

Benefícios da observabilidade

Os sistemas observáveis são mais fáceis de entender, controlar e consertar. À medida que os sistemas mudam – seja de propósito porque você está implantando um novo software, uma nova configuração, aumentando/diminuindo a escala, ou talvez por causa de alguma outra ação desconhecida – a observabilidade permite que os desenvolvedores entendam quando o sistema começa a não estar no estado pretendido.

As ferramentas modernas de observabilidade podem identificar automaticamente vários problemas e suas causas, como falhas causadas por alterações de rotina, regressões que afetam apenas clientes específicos e erros de downstream em serviços e SaaS (Software as a Service) de terceiros.

Benefícios de observabilidade por função:

Desenvolvedores

A observabilidade reduz a quantidade de estresse ao implantar o código ou fazer alterações no sistema. Ao destacar “o que mudou” após qualquer implantação ou alerta sobre valores discrepantes, os problemas que afetam o cliente podem ser encontrados rapidamente e revertidos antes que os SLOs sejam interrompidos.

Ao revelar o caminho crítico das solicitações de ponta a ponta e apresentar apenas os dados





relevantes para resolver um problema, a observabilidade permite melhores fluxos de trabalho para depuração, otimização de desempenho e combate a incêndios.

Descartando sinais que provavelmente não contribuíram para a causa raiz, os desenvolvedores podem formar e investigar hipóteses mais eficazes.

Equipes

Para equipes de todos os tamanhos, a observabilidade oferece uma visão compartilhada do sistema. Isso inclui sua integridade, sua arquitetura, seu desempenho ao longo do tempo e como as solicitações passam de front-end/aplicativos da Web para serviços de back-end e de terceiros.

A observabilidade fornece contexto entre funções e organizações, pois permite que desenvolvedores, operadores, gerentes, PMs e quaisquer outros membros da equipe trabalhem com as mesmas visualizações e insights sobre serviços, clientes específicos, consultas SQL etc.

Como as ferramentas de observabilidade permitem a captura automatizada de qualquer momento no tempo, elas servem como um registro histórico da arquitetura do sistema, dependências e integridade do serviço — o que foi e o que mudou ao longo do tempo.

A observabilidade gera post-mortems mais eficazes após os incidentes, porque permite que a equipe revise o comportamento real do sistema no momento do incidente, em vez de confiar nas lembranças de indivíduos que operam sob uma situação estressante.

Negócios

Em última análise, ao tornar um sistema observável, as organizações podem liberar mais código, com mais rapidez e segurança.

O que geralmente determina se sua empresa é bem-sucedida ou não é a capacidade de mudar





e enviar novos recursos. Mas essa mudança é diretamente oposta à estabilidade de seus sistemas. Isso gera uma tensão básica: você precisa mudar para poder expandir seus negócios, mas sempre que uma mudança é introduzida, o risco é introduzido junto, podendo criar resultados negativos em um senso de negócios.

A observabilidade resolve essa tensão básica, ajudando as empresas a poderem realizar mudanças com níveis mais altos de confiança e descobrir se estão ou não tendo o efeito pretendido e assim, limitar o impacto negativo da mudança.

O resultado são lançamentos mais confiantes em uma velocidade mais alta. Você pode implantar com mais frequência e com mais confiança, pois possui ferramentas que o ajudarão a entender o que está errado, e o auxiliam a isolar quaisquer problemas, e por fim a realizar as melhorias imediatas necessárias.

Em última análise, ele permite que você mantenha os clientes satisfeitos com menos tempo de inatividade, novos recursos e sistemas mais rápidos.

Os três pilares da observabilidade

Historicamente, os três tipos de telemetria são chamados de “três pilares” da observabilidade: tipos de dados separados geralmente com seus próprios painéis.

A crescente escala e complexidade do software levou à mudanças nesse modelo, no entanto, os profissionais não apenas identificaram as inter-relações entre esses tipos de dados de telemetria, mas também os fluxos de trabalho coordenados que os envolvem.

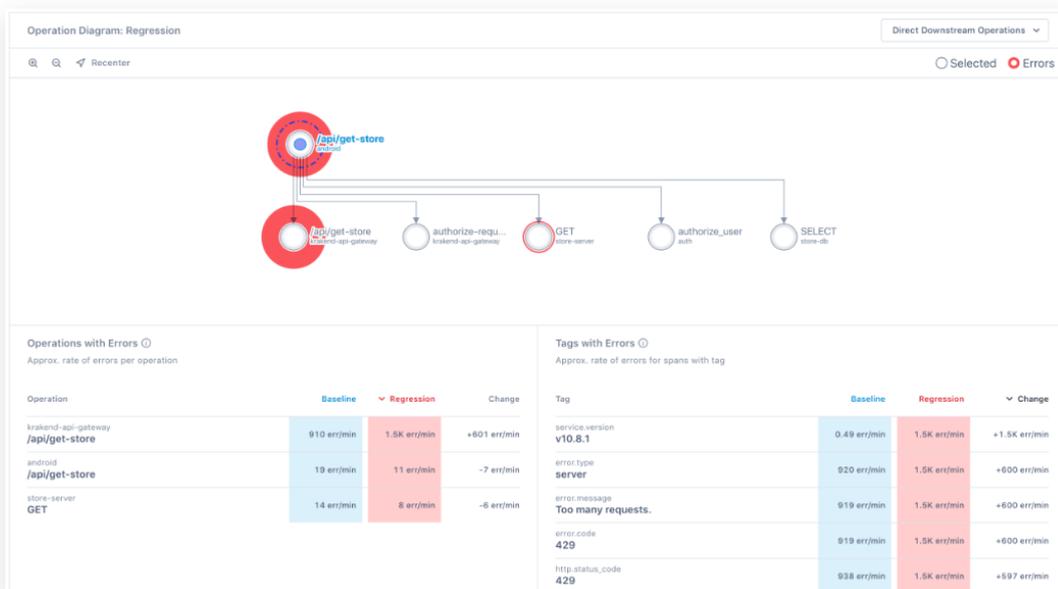
Por exemplo, painéis de métricas de séries temporais podem ser usados para identificar um subconjunto de rastreamentos que apontam para problemas ou bugs subjacentes — e as mensagens de log associadas a esses rastreamentos podem identificar a causa raiz do problema. Em seguida, novas métricas podem ser configuradas para identificar de forma mais proativa problemas semelhantes antes do próximo incidente.





Além disso, quando visualizados de forma agregada, os rastreamentos podem revelar insights imediatos sobre o que está causando o maior impacto no desempenho ou na experiência do cliente e mostrar apenas as métricas e os logs relevantes para um problema.

Digamos que haja uma regressão repentina no desempenho de um serviço de back-end específico, no fundo de sua pilha. Acontece que o problema subjacente era que um de seus muitos clientes, mudou o seu padrão de tráfego e começou a enviar solicitações significativamente mais complexas. Isso se torna óbvio em segundos depois de analisar as estatísticas de rastreamento agregadas, algo que levaria dias se fosse analisado apenas logs, métricas e rastreamentos individuais por conta própria.



Resumindo: observabilidade não é simplesmente telemetria – é como essa telemetria é usada para resolver problemas e, em última análise, criar uma experiência melhor para os clientes.





Ferramentas de observabilidade: regras que não podem ser quebradas

Existem três formas principais de fazer um sistema distribuído observável. Como qualquer escolha, cada uma vem com um conjunto de benefícios e custos (o qual pode ser literal).

As equipes podem criar suas próprias ferramentas de observabilidade, trabalhar com software de código aberto, como Jaeger ou Zipkin, ou adquirir uma solução de observabilidade.

Independentemente de como você decide tornar seu sistema observável, há algumas regras verificadas que se aplicam a todas as soluções de observabilidade.

As ferramentas de observabilidade devem ser simples de integrar

Se a integração for muito difícil, é improvável que seu projeto saia do papel. Quem tem ciclos extras para projetos e testes de integração de vários meses? E, mesmo que haja alguma disponibilidade, quem quer assumir esse tipo de projeto?

Esforços de observabilidade bem-sucedidos se conectam às ferramentas que você já está usando. Eles devem ser capazes de oferecer suporte a ambientes políglotas (com quaisquer linguagens e estruturas que você use), integrar-se à sua malha de serviço ou plataforma de contêiner e conectar-se ao Slack e ao PagerDuty ou a qualquer sistema que sua equipe de chamada prefira.

As ferramentas de observabilidade devem ser fáceis de usar

Se a plataforma for muito difícil de aprender ou usar diariamente, ela não fará parte dos processos e fluxos de trabalho existentes. Os desenvolvedores não se sentirão confortáveis em recorrer à ferramenta durante momentos de alto estresse, e pouca melhoria será feita na saúde e confiabilidade do sistema.





As ferramentas de observabilidade devem ser construídas em telemetria de alta qualidade

Amostragem inicial, amostragem aleatória e a maioria dos métodos de amostragem de dados simplesmente não permitem a observabilidade em escala. Dados de alta fidelidade são necessários para identificar discrepâncias ou problemas específicos em um sistema distribuído, pois a resolução de incidentes pode exigir a análise de eventos intermitentes, infrequentes e raros.

As ferramentas de observabilidade devem ser em tempo real

Relatórios, painéis e consultas precisam fornecer informações sobre o que está acontecendo “agora”, para que os desenvolvedores possam entender a gravidade de um problema ou o impacto imediato de suas otimizações de desempenho. Ao depurar um problema ou combater uma interrupção, as informações de dez minutos atrás simplesmente não serão suficientes.

As ferramentas de observabilidade devem agregar e visualizar seus dados

Sistemas em escala produzem uma enorme quantidade de dados e em muito mais quantidade do que pode ser facilmente compreendido por humanos, sem algum tipo de orientação. Para que uma ferramenta de observabilidade seja eficaz, ela precisa tornar os insights trazidos por esses dados óbvios. Isso inclui resumos visuais interativos para resolução de incidentes e painéis claros que oferecem uma compreensão rápida de qualquer evento.

As ferramentas de observabilidade devem fornecer um contexto irrefutável

O que é uma investigação sem contexto? Adivinhação? Tentativa e erro? Uma ferramenta de observabilidade deve orientar seus usuários para a resolução de incidentes bem-sucedidos e a exploração do sistema, fornecendo contexto a cada passo do caminho.





Contexto temporal: como algo parece agora vs. uma hora, um dia ou uma semana atrás? Como era isso há 3 meses? Como era antes da implantação?

Contexto relativo: Quanto isso mudou em relação a outras mudanças no sistema?

Contexto relacional: O que depende disso e do que depende? Como as mudanças nessa cadeia de dependência afetarão outros serviços?

Contexto proporcional: Qual é o escopo de um incidente ou problema? Quantos clientes, versões ou geografias são afetados? Os clientes VIP são mais ou menos impactados?

As ferramentas de observabilidade devem fornecer ROI para o negócio

Isso pode parecer óbvio, mas pode ser muito fácil confundir involuntariamente como uma ferramenta é percebida por um desenvolvedor e como ela realmente gera ou não gera valor comercial. Em última análise, as ferramentas de observabilidade devem melhorar a experiência do cliente, aumentar a velocidade do desenvolvedor e garantir um sistema mais confiável, resiliente e estável em escala.





Sobre a OpServices

Empresa especializada em **monitoração avançada e visualização de dados personalizadas**, com 20 anos de mercado. É reconhecida por desenvolver soluções e serviços orientados às necessidades de processos de negócios específicas para cada cliente.

Contato

(51) 3275.3588

vendas@opservices.com

www.opservices.com.br

