Backtrace - A Platform for Software Debugging and Crash Reporting

The Backtrace Vision

Backtrace was founded with a vision to build the best cross-platform, native application crash and error reporting technology for today's complex software, video game systems, embedded technology, and IoT devices. The software development teams that work in these environments now have a greater impact on how the company performs. When software fails to perform consistently, customers will disappear and employees will function less efficiently.

Backtrace developed a platform that empowers organizations to make new and better choices for investing in software. The Backtrace platform has broken new ground with technology that:

A. Captures and analyzes crash data from multiple platforms, including desktop (Windows, Mac), mobile (iOS, Android) server systems (Linux), embedded devices (Linux, RTOS), and video game consoles, with new levels of granularity.
B. Provides teams with easy to use analytics to prioritize bugs and application failures based on their impact.
C. Enables collaboration and integration with the tools you have in place today to speed ability to resolve the issue.

The Backtrace Platform

Backtrace gives your team the automation and diagnostic tools they need to spot errors that matter, understand their impact, explore the context, and zero in on causes in hours or minutes instead of days or weeks.

Backtrace is a turn-key solution that was purpose-built for crash and exception analysis. This is distinct from homegrown solutions or log analysis tools that were not designed for debugging:

- Limited (or no) call stack symbolication for native application crashes
- Do not provide causal clues based on debugger analysis to support root cause investigation
- Rudimentary deduplication implies more noise and less reliable identification of uniqueness of an error

Backtrace was originally built to support aggregation and analysis of Linux core dumps from C/C++ based applications. Backtrace has expanded to support other OSes like Windows, macOS, iOS, Android, and various distributions of Linux, as well as additional formats and languages including minidumps, Swift & iOS, Java and Kotlin, C#, Electron, Go, Node, JavaScript, and Python

The core tenants of the platform streamline the capture, analysis and resolution flow of errors from your applications or infrastructure.

Capture

Backtrace offers various methods for crash or fault capture. Each of them was designed to be high performance with low impact to running code. Integration with your software is intended to be simple and easy, with uncomplicated code changes or an OS-level configuration to take a snapshot and send crash reports to our servers.

Backtrace's 'coresnap' tool automatically manages all core dumps on your Linux and FreeBSD systems, including archival, aggregation and analysis. Backtrace also has native support for the minidump format, seamlessly integrating with the open-source Crashpad and Breakpad libraries, and offering native support for the Electron CrashReporter. Backtrace also has reporting libraries for languages like Swift, Java/Kotlin (for Android), Go, Node, Javascript, Python, and .NET/C#. An HTTP based API and JSON format allow submission of error data from platforms that are not supported out of the box.

A key feature of the Backtrace system is it's first-class attribute/metadata system. Backtrace attributes allow for the capture of additional context alongside error data. Reporting libraries will automatically gather commonly used system data (OS, process uptime, memory usage, etc.), and allow for the addition of custom attributes, giving you the ability to capture the context that matters to you. Examples of custom attributes include app version, build tag, screen resolution, hardware revision, anonymized user id, and more.

Attributes are indexed, allowing for custom grouping, searching, filtering based on any attribute via the Backtrace analytics engine. Attributes serve as the basis for custom reports and our powerful triage and prioritization functionality.

Analyze

At the heart of the Backtrace platform is a fit-for-purpose, custom developed database called CRDB. CRDB is a columnar database that indexes and provides searchability for all information and attributes collected about crashes.

Backtrace's web console and CLI (morgue) provide a query language that makes it easy for you to search, group by, and aggregate various data points to provide you with powerful analytics on top of your error and crash data. You can quickly slice and dice data in a way that makes sense for your business needs.

The web console allows for the quick visualization of trends across attributes and groups of errors. The analysis engine allows for nested selection and hierarchical joins, allowing you to answer such questions as "When was the first time this user has seen an error?" or "When was this error first introduced?" The web console also provides a crash debugger UI to view the submitted dumps and detailed call stacks from the convenience of your web browser. The debugger provides a complete view of the error report, including system stats, threads, call stacks, registers, attributes and annotations. If available, The debugger will indicate the thread that crashed and the frame that is most likely to be relevant to the crash.

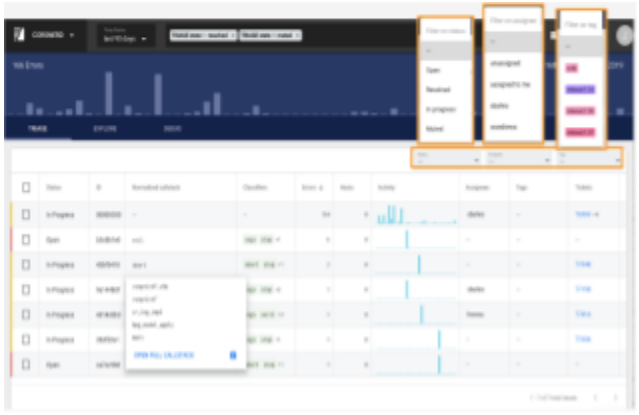

Figure 1 - Triage and Prioritize the most important errors with intelligent deduplication, automated symbolication, sophisticated filter and search, 2 way sync with issue tracking systems like Jira, and more.

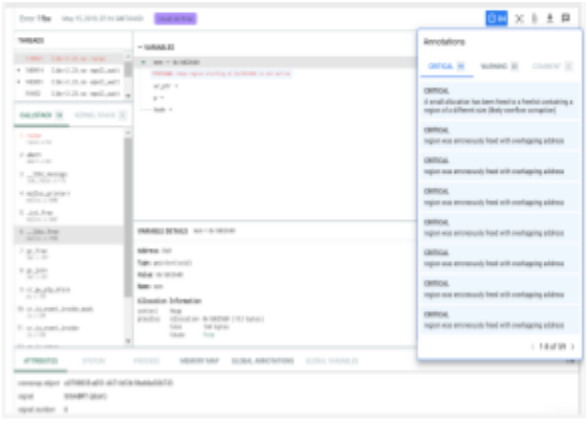

Figure 2 - Use the Debugger to view full thread, call stack, and contextual details from the convenience of your browser. See variable values, jump to faulting frame, view indicators of anomalous behaviors, and more.

A purpose-built debugger (ptrace) for UNIX and UNIX-like systems built from the ground-up provides advanced analysis of application state during errors. Backtrace provides automated classification to give you a head start on root cause analysis. Backtrace will identify issues like null dereference, floating point error, heap corruption, divide by zero, stack overflow and more. In addition, intelligent deduplication techniques makes it easier to identify common issues. Finally, Backtrace also offers advanced symbol management that automates much of what is needed to provide human readable call stacks for improved debuggability.

For more powerful, custom analysis and the ability to export error data to other systems, Backtrace includes a terminal client and HTTP APIs that directly leverage our purpose-built database. Backtrace, unlike other systems, allows for full flexibility of analysis on the complete error set and to export that data in the JSON format for seamlessly integration with other data stores (e.g. user-data, session data stores).

Resolve

Backtrace's web interface includes built-in triage and prioritization systems that seamlessly integrate into third-party tools. Synchronization capabilities allow for work management state such as assignee and status to stay up to date across systems. For example, if you close a JIRA ticket tied to a group in Backtrace and a new error report comes in to that group, Backtrace can automatically re-open that JIRA ticket. Backtrace also integrates into other third-party systems such as messaging apps, Amazon's SNS, and generic web services. The system can trigger actions in third-party systems based on events observed by Backtrace, increasing productivity by

embedding relevant context into those systems that you use to work. For example, some Backtrace users send a Slack or Webex Teams message for every instance of a crash that is seen, or decide to create a new ticket in their issue tracking system the first time a crash is seen.

Following is a list of the types of integrations Backtrace includes to support your resolution procedures:

- Messaging - Slack, MS Teams, Cisco Webex Teams, Email & IRC. Support for legacy products including Atlassian Hipchat & Stride
- Issue Tracking - Jira, GitHub Issues, Phabricator, FogBugz, Asana
- Alerts / Operations - VictorOps, PagerDuty, OpsGenie
- Monitoring - Data Dog, Circonus
- Queues - Amazon SNS, SQS
- Custom - HTTP Webhooks

Beyond these integrations, we also have customers who use Backtrace as a central location to store and retrieve (or download) their minidump files when they need them for debugging in another client like Visual Studio.

Comparing Backtrace with Generic Error Monitoring Tools

As organizations look at alternatives to Backtrace, they'll quickly find key capabilities that are unique to Backtrace and provide our users with improved time to discover and resolve application issues.

1. Query and Group By - Practitioners and managers alike want to search all their call stacks and error data, and group by different attributes such as User, Version, or any other custom attribute. They want to treat their error data like any other BI data that they can report and action on. Only Backtrace offers this capability.
2. Call stack and Deduplication accuracy - We hear from users that Backtrace's call stack unwinding and deduplication techniques provide for more accurate grouping of crashes. We are constantly adding rules and support for new libraries, and are confident that we can group minidump crashes better than market alternatives that we've seen.
3. Retention and management of the crash/exception artifacts and metadata - Our customers want to control how long they retain artifacts like minidump files for, and who can download them for external analysis when needed. They want to further retain historical metadata for longer periods of time so they can perform longer term trend analysis, Only Backtrace offers this capability.

4. Deployment Flexibility - Large enterprises want to control where and how they manage their data. They want options beyond standard multi-tenant SaaS deployments, such that they can choose hosted single tenants, dedicated servers, or on-premises deployments. Only Backtrace offers customers fully flexible deployment options, so customers are in full control of where their data lives and how it is co-mingled with other's data.
5. Breadth of Platforms Supported - While web and mobile applications have seen large growth in recent years, desktop apps, consumer devices, video game consoles, and server applications continue to offer a wide range of locations where your native code could be executing and crashing. Development teams want one platform that can manage crash reports from all these runtimes. Only Backtrace provides true cross-platform support for native and web crash and error reporting.

Backtrace has developed a more robust platform to address the debugging needs of software organizations that develop desktop, mobile, server, or web based applications at large enterprises. What follows is a further explanation of the advantages of Backtrace compared to other alternative solutions.

Backtrace Advantages

- True cross platform support for native and web applications.
    - Desktops - Windows and macOS, including minidump and other crash artifacts.
    - Mobile - iOS and Android, including system attributes around device, screen, WiFi / bluetooth / GPS, etc.
    - Servers - Linux variants, FreeBSD, Apache Traffic Server, and more.
    - Video Game Consoles like Xbox One and PS4
    - Set Top Boxes and other IoT devices using the open source RDK platform, Linux, RTOS, or other.
    - Web - Error and crash reporting from JavaScript and .NET, Python, Ruby, Go
- Sophisticated deduplication
    - Deduplication is the method used by the platform to identify similar crashes that should have a common root cause.
        - Backtrace uses semantic analysis, faulting function matching, heuristic matching and more to better group similar crashes.
        - Other platforms typically use simplistic callstack matching with considerations for line numbers, a method Backtrace also utilizes.
- Advanced data analysis capabilities

    - Practitioners and managers alike want to search all their call stacks and error data, and group by different attributes such as playerID, device type, wifi connectivity status, or any other custom attribute.
    - Backtrace provides the functionality to group, filter, search on any reported attribute or information in a call stack.
    - Backtrace supports comparison operations such as regular expressions, substring matches, equivalence, and their respective inverses.
    - Our generic export API also allows for custom visualizations such as flame graphs, histograms, and more, and provides ability to load diagnostic data into third party tools.
- Full-featured symbol management ensures more callstacks have the symbols they need so users can interpret the cause of a native crash.
    - Comprehensive symbol format support, including .pdb, .sym, dSYM, ELF, DWARF.
    - Symbol Server for on-demand retrieval of 3rd party dependent libraries (i.e. Apple, Electron, Microsoft, and Mozilla) or your private symbol server.
    - Notification of Missing Symbols and Symbol Upload wizard.
    - Binary symbol format significantly speeds processing and loading. (Customers are seeing up to a 30x improvement in reprocessing time and debugger load times with this enhancement).
- PII Data Scrubbers: Backtrace has a far more powerful data scrubbing processor to remove PII data from submissions.
    - Built in scrubbers for credit card and social security numbers, as well as encryption key or environment variables.
    - Regex can be used to create additional scrubbers (i.e. IP Address can be scrubbed with regexp='[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+').
    - System will scrub submitted dump file, including execution path, memory, environment variables, register values and user defined attributes.
- Full-Featured data export API
    - Instead of relying on web-hook and queues, Backtrace provides an HTTP API through HTTP to query and export data when you need it.
    - The HTTP API provides the full features of the analytical engine, allowing you to leverage Backtrace's nested selections and hierarchical joins to reduce your up-front work for cross-system data analysis.
- Optimized for Performance and Scale
    - Backtrace provides much faster query and load times for crash dump analysis. This is due to the Backtrace fit for purpose database that provides extremely

efficient query performance, and storage with optimized formats and intelligent caching to make loading dump files for analysis more efficient.
    - Backtrace has been proven to handle crash rates at the 100Ms, with intelligent sampling and retention policies.
    - Backtrace sampling allows for the retention of attributes without storing the full error report. Other solutions do not provide this functionality.
- Generic HTTP error reporting API
    - Backtrace provides the ability to manage errors from unsupported platforms through our generic HTTP API.

Feature & Capabilities List

Feature	Capabilities
Platforms and Languages Supported	Native core dumps (Linux, FreeBSD) Minidumps (Windows, Mac, Linux) Game Consoles - Xbox One, PS, Game Engines - UE4, Unity Devices - Set Top Boxes (RDK), IoT and other Embedded Languages - C/C++, C#/.NET, Electron, JavaScript, Node, Java, Kotlin, Python, Go, Rust, Swift, Objective-C, Custom JSON.
Deployment Flexibility	**Customer Choice** - On Premises, Dedicated Host, Shared Host
Metadata and Attachments	System runtime data (OS, Process uptime, Memory usage, battery level, etc); Custom Attributes; Logs, File attachments alongside an error
Debug Symbol Management	Formats: pdb, dSYM, ELF, sym, ProGuard, SourceMaps Advanced Features: On-demand retrieval of 3rd party libraries or your private symbol server; Reprocessing capabilities; Binary symbol format improves processing speeds up to 30x.
Query with Group, Search & Sort	**Full featured query engine** includes “Group By” any metadata, **sophisticated filter operators**, **searching the callstack**, and more.
Deduplication & Grouping	**Custom rules based system** including faulting function and heuristic matching
Lifecycle Management	Lifecycle States (Open, In Progress, Closed, Muted), Assignee, Comments, 2 way sync with Jira and other Issue tracking systems
Misc	PII Data Scrubbing on Minidump files, **Minidump download**, Custom Retention Policies, Full Sampling Control, **Full featured Query API**