How OverOps Helped Zynga Gain Insights About Errors and Exceptions

Zynga (NASDAQ: ZNGA) is a leading developer of the world's most popular social games that are played by millions of people around the world each day.

zynga

Production Monitoring Ecosystem

OverOps splunk> Nagios DATADOG HipChat

Key challenges

Zynga had a hard time finding and debugging exceptions, even for their stable applications, due to the massive amount of lines written to the log. Developers needed to sift through logs in hopes of finding the specific issue they had encountered. They needed a tool to help them debug in production.

1.7k

Employees

$764M

Revenue

Error Resolution (Pre-OverOps)

There's only some information that fits inside a log line, and it gives a narrow view of what's going on inside the application. As a result, the engineering team had no way of knowing that certain services were failing since there was no logging for uncaught exceptions. The company needed a tool that could help production debugging.

Error Resolution (With OverOps)

Now with OverOps, Zynga catches all uncaught exceptions including the complete source code and variable state, across the entire call stack. Zynga's engineers can now immediately identify critical errors, and see the exact variable state that caused them.

Avg. Time to Resolve Production Errors

3 Days → 5 Minutes

99% Reduction ✓

RJ Lim, Senior Software Engineer at Zynga:
"Some problems are very hard to debug even if you have logs. With OverOps we have all of the information we need."

Why OverOps for Zynga

1 Variable values across the entire call stack for every production error

2 Identifies root cause for each error without sifting through logs

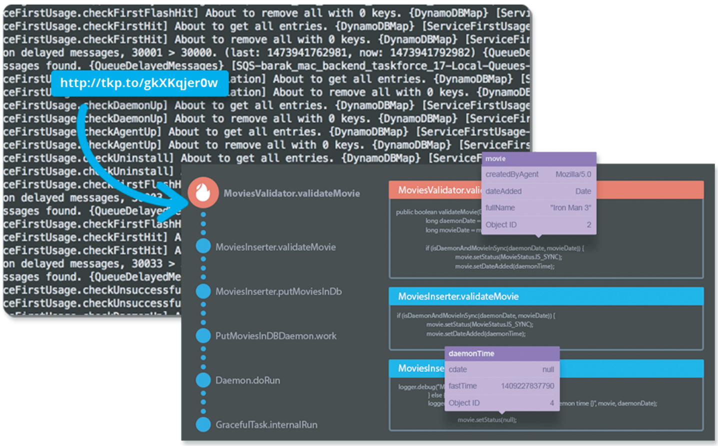
3 Helps diagnose severity level for each error

4 Delivers proactive approach for fixing errors instead of waiting for them to happen

"OverOps took out the guesswork and helped us diagnose the problem quickly and fix it right away."

Zynga uses OverOps to detect errors and exceptions within their applications, and also to tidy up the logs themselves and make them meaningful. Instead of ignoring millions of error lines, Zynga can now know what's important, get real-time notifications via HipChat and fix issues as soon as they occur.

Schedule a demo with an OverOps monitoring engineer or start your **free 14 day trial today**



```
ceFirstUsage.checkFirstFlashHit] About to remove all with 0 keys. {DynamoDBMap} [Servi
ceFirstUsage.checkFirstHit] About to get all entries. {DynamoDBMap} [ServiceFirstUsage
ceFirstUsage.checkFirstHit] About to remove all with 0 keys. {DynamoDBMap} [ServiceFir
on delayed messages, 30001 > 30000. (last: 1473941762981, now: 1473941792982) {QueueDe
ssages found. {QueueDelayedMessages} [SQS-barak_mac_backend_taskforce_17-Local-Queues-
ceFirstU                            ation] About to get all entries. {DynamoDBMap} [S
ceFirstU                            ation] About to remove all with 0 keys. {DynamoDB
ceFirstUsage.checkDaemonUp] About to get all entries. {DynamoDBMap} [ServiceFirstUsage
ceFirstUsage.checkDaemonUp] About to remove all with 0 keys. {DynamoDBMap} [ServiceFir
ceFirstUsage.checkAgentUp] About to get all entries. {DynamoDBMap} [ServiceFirstUsage-
ceFirstUsage.checkAgentUp] About to remove all with 0 keys. {DynamoDBMap} [ServiceFirs
ceFirstUsage.checkUninstall] About to get all entries. {DynamoDBMap} [ServiceF
ceFirstUsage.checkUninstall]
ceFirstUsage.checkFirstFlashH
on delayed messages,
ssages found. {QueueDelaye
ceFirstUsage.checkFirstFlashH
ceFirstUsage.checkFirstHit] A
ceFirstUsage.checkFirstHit] A
on delayed messages, 30033 >
ssages found. {QueueDelayedMe
ceFirstUsage.checkUnsuccessfu
ceFirstUsage.checkUnsuccessfu
```

http://tkp.to/gkXKqjerOw

```
public boolean validateMovie(
        long daemonDate =
        long movieDate = m

        if (isDaemonAndMovieInSync(daemonDate, movieDate)) {
                movie.setStatus(MovieStatus.IS_SYNC);
                movie.setDateAdded(daemonTime);
```

```
if (isDaemonAndMovieInSync(daemonDate, movieDate)) {
        movie.setStatus(MovieStatus.IS_SYNC);
        movie.setDateAdded(daemonTime);
```

```
logger.debug("M
        } else
        logge                    emon time {}", movie, daemonDate);

                movie.setStatus(null);
```