

BITGRAVITY PROVIDES HD QUALITY LIVE STREAMING SERVICES WITH ITS BG LIVE OFFERING.

BitGravity is based in Burlingame, CA, USA, and is a wholly owned subsidiary of Tata Communications, Ltd.



CUSTOMER PROFILE

Founded in 2006, BitGravity launched the world's first content delivery network built and optimized for on-demand video delivery and HD live streaming. Since then, they've broadened their capabilities and each day thousands of organizations around the globe use BitGravity products to accelerate Web sites, serve better performing videos, and reliably stream events and 24/7 broadcasts online in HD.

BUSINESS NEED

BitGravity provides HD quality live streaming services with its BG Live offering. This product provides customers with a way to stream high quality live events around the world to upwards of hundreds of thousands of simultaneous viewers in over 200 countries and territories. The technical staff at BitGravity had identified an inefficiency in the underlying FreeBSD code that was increasing processing demands in regard to vector data. The current way of writing a vector of data to a vector of file descriptors is inefficient due to the amount of data needing to buffer between user space (applications) and kernel space (the native operating system). On the scale at which BG Live performs, this problem created significant inefficiencies in vector data writes, increasing the load on BitGravity's hardware infrastructure thereby increasing costs.

SOLUTION

BitGravity came to iXsystems Inc. for their specialized FreeBSD knowledge in order to develop a new system call to reduce this inefficiency, thereby improving their product performance and allowing them to better serve their BG Live customers at reduced costs.

RESULTS

Manifold increase in vector write efficiency in pre-integration testing environment. Integration of system call led to new algorithms to push the performance of BitGravity servers to very high levels beyond original expectations. This work also led to an unforeseen benefit, helping to distribute kernel interrupts amongst the available CPU cores, allowing for more concurrent connections to their BG Live service.

IN DEPTH REVIEW:

PROBLEM:

The writev system call is a method for writing vector data to a file descriptor in Unix based operating systems. The writev command “writes data to file descriptor (fd), and from the buffers described by vector. The number of buffers is specified by count (fdcnt). The buffers are used in the order specified. [It] Operates just like write, except that data is taken from vector instead of a contiguous buffer” <http://linux.about.com/library/cmd/blcmdl2_writev.htm>. The problem is that the Operating system segregates virtual memory between the kernel and userland, creating significant inefficiency due to the large number of swaps between user space and kernel space.

SOLUTION DESCRIPTION:

Create a new system call, writev which can be used to write a vector of data to a vector of file descriptors efficiently. This system call should incur the cost of copying data from kernel space to user space only once, rather than multiple times over.

The writev system call is equivalent to calling writev with the same arguments once for every file descriptor contained in file descriptors (fds). The fdcnt parameter indicates the length of fds. The returns parameter should point to an array of length fdcnt, which will then receive the list of return codes corresponding to the writev calls. The errors parameter should point to an array of length fdcnt, which shall receive the list of errors corresponding to the what the individual writev calls would have returned.

PRELIMINARY TESTING RESULTS:

```
~/bitgravity % time ./testsuite/writevtest/writevtest
./testsuite/writevtest/writevtest 0.02s user 5.66s system 99% cpu 5.678 total
~/bitgravity % time ./testsuite/writevtest/writevtest -U
./testsuite/writevtest/writevtest -U 0.22s user 35.19s system 99% cpu 35.410 total
```

IMPLEMENTATION:

In order to implement this solution, BitGravity came to iXsystems to employ their Professional Development Services. iXsystems then proceeded to work with BitGravity's technical staff to analyze their needs develop a working solution and assigning the FreeBSD developer most well-suited to the task. From there, an agreement was reached and implementation of the solution was begun. Upon initial code completion, preintegration testing of the syscall itself showed promising results, providing a 623% increase in vector write performance.

As development work progressed, additional needs became apparent and iXsystems worked with BitGravity to come to timely resolutions to those matters. For example, it was also deemed important to improve the syscall output to give more information as to the number of write failures. iXsystems then promptly held a meeting with the BitGravity staff and worked out a solution to achieve these results in an efficient fashion.

From this point, the code was finalized and sent to BitGravity for their technical team to integrate with their application. This work not only provided the desired outcome of vector write performance gains, but also led to additional benefits.

“As part of integrating the changes we found other bottlenecks in the application and as a result have come up with new algorithms to push the performance of the servers to very high levels - much more than what other folks here expected. So I am very pleased with the results.”

-Samik Mukherjee, BitGravity Project Manager

Even beyond this, the integration process partially resolved another issue their technical team was having with uneven distribution of kernel interrupts. This outcome will allow BitGravity to serve more concurrent viewers at any given time, thereby improving their ability to serve their customers with better website acceleration and video performance and more reliable event streaming capabilities.

IMPLEMENTATION:

iXsystems provided BitGravity with timely assistance and specialized FreeBSD expertise suited to their needs. The services rendered led to greater than expected performance gains and also led to unanticipated benefits by partially solving another problem the BitGravity team had been working on. This code implementation will undoubtedly help BitGravity better serve their customers with their BG Live service and reduce their internal costs to do so in the process.