POWERING CHANGE WITH AI | NVIDIA

USING AI TO REVEAL INSIGHTS IN ENTERPRISE LOG FILE DATA

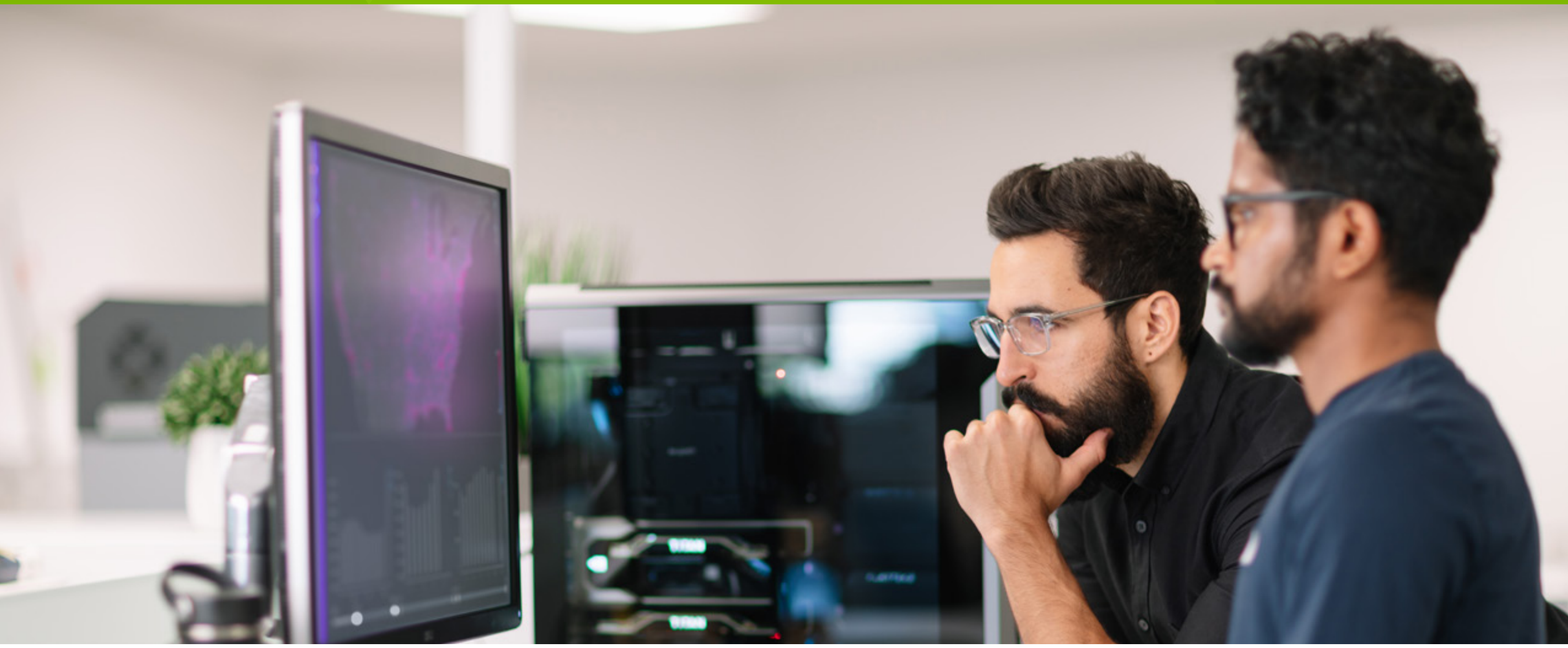


Image courtesy of NVIDIA

NVIDIA DGX Systems and AI help businesses zero in on problems before they surface.

INTRODUCTION

Businesses in every industry run on processes that generate oceans of log file data. Modern enterprises spend inordinate amounts of time and resources scouring machine or process log files to inspect error conditions, which often include false positives. This drives up costs and reduces agility in anticipating and responding to actual business-impacting conditions.

Enterprises are leveraging AI-based log analysis to prevent these costly mistakes: Security teams can analyze logs for anomalies and prevent breaches before they happen; manufacturing departments can detect failures before they spread to the next part of an assembly line and purchase parts before they wear down, reducing unplanned downtime; and IT operations—bombarded with logs from all aspects of the system, including network, web server, and database—can now troubleshoot performance issues before they become problems. In some industries like financial services, this could save millions of dollars, as network issues and unwanted latency can delay the completion of trades, causing missed arbitrage opportunities. At the same time, AI-based log analysis is being used to uncover missed revenue opportunities by analyzing customer transaction logs in real time to offer upsell opportunities or incentives that prevent transactions from being abandoned.

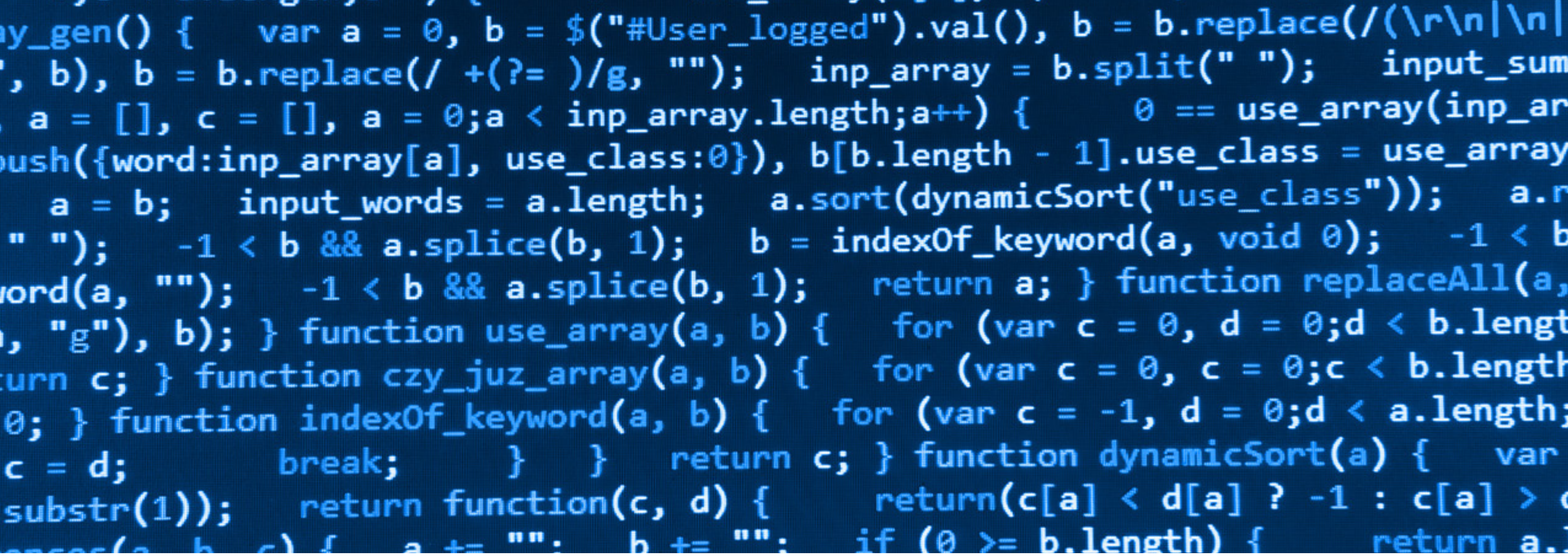
AI and deep learning are invaluable tools that can help organizations derive real-time insights from log file data. These insights can lead to increased productivity, fewer false positives, and faster responses to problems before they surface. In this case study, we explore how NVIDIA has implemented AI-powered log file analysis to streamline the development of the world's most advanced GPUs.

CHALLENGE

Designing the GPU is a very complicated task. NVIDIA engineers must ensure that the connections between each and every functional block—including billions of transistors—work properly. Circuit verification and emulation occur to test the functional blocks, with each block undergoing thousands of tests.

Often, the log files generated by these tests have their own cryptic language, and overreporting on errors is common; the word "error" will appear many times in the file, even when the events are completely benign, resulting in many false positives.

In physical design, the design process kicks off thousands of runs every day; an average of 15 percent of those runs fail for various reasons. That can amount to thousands of failed log files that a team of engineers have to go through—and each file can have millions of lines of code. For an experienced engineer, it typically takes ten to fifteen minutes to figure out the problem. In many cases, it can take much longer. Just five minutes of debugging time saved for each log file would save well over a hundred engineering hours per day, or more than two million dollars in operating expenses per year.



```
y_gen() {    var a = 0, b = $("#User_logged").val(), b = b.replace(/(\r\n|\n|
', b), b = b.replace(/ +(?= )/g, "");    inp_array = b.split(" ");    input_sum
 a = [], c = [], a = 0;a < inp_array.length;a++) {      0 == use_array(inp_ar
oush({word:inp_array[a], use_class:0}), b[b.length - 1].use_class = use_array
   a = b;    input_words = a.length;    a.sort(dynamicSort("use_class"));    a.r
" ");    -1 < b && a.splice(b, 1);    b = indexOf_keyword(a, void 0);    -1 < b
ord(a, "");    -1 < b && a.splice(b, 1);    return a; } function replaceAll(a,
, "g"), b); } function use_array(a, b) {    for (var c = 0, d = 0;d < b.lengt
urn c; } function czy_juz_array(a, b) {    for (var c = 0, c = 0;c < b.length
0; } function indexOf_keyword(a, b) {    for (var c = -1, d = 0;d < a.length;
c = d;          break;       }    }    return c; } function dynamicSort(a) {    var
substr(1));    return function(c, d) {      return(c[a] < d[a] ? -1 : c[a] > 
```

Image courtesy of NVIDIA

SUMMARY

- Millions of dollars are spent in the physical design of graphics architectures, generating thousands of failed log files each day.
- Engineers spend ten to fifteen minutes per failed log file just to understand the critical error and where to start debugging.
- Using a deep learning framework powered by NVIDIA DGX-1™, NVIDIA was able to cut debugging time and save over a hundred engineering hours per day, resulting in more than two million dollars saved per year.
- This same model and technique for analyzing log files can be applied to other departments in the company, including IT operations, security, and testing.

INFRASTRUCTURE

Model training: NVIDIA DGX-1

Production inference: NVIDIA V100 Tensor Core GPUs

SOLUTION

NVIDIA's design team turned to deep learning to solve their problem. By comparing successful and failed log files, they could gather reasons for failure from hundreds of thousands of log files or warning messages. But it wasn't as simple as finding keywords like "error," "crash," or "fatal." In many instances, lines with these keywords would still pass and be considered benign. The failed logs would contain additional messages that would have a higher chance of revealing the root cause of the failures. The team sought to find a way to extract these anomalous lines automatically and took inspiration from the recent developments in natural language processing (NLP) text-classification methods.

The team had already collected the data from thousands of sets of passed and failed log files. The next step was to prepare the data to be fed into a deep learning model. This involved filtering and simplifying the lines (removing data like time stamps, instance names, and users). From here, the team was able to feed the training dataset, which consisted of many log file sets, both successful and failed, into a seq2seq autoencoder framework to learn the grammar of the lines. They performed tokenized differentiation on the failed logs against passed logs to classify each line as benign or anomalous.

The NVIDIA DGX-1 was used to train the seq2seq autoencoder for anomaly detection. DGX-1 is an efficient supercomputer for AI and deep learning that delivers one petaFLOPS of compute performance in a single node. "The team was able to train their model to 95 percent accuracy. They wanted to have a higher recall rate and didn't mind having a few false positives to ensure all errors were caught," says NVIDIA's principal engineer in deep learning applications.

"We've significantly reduced debugging time and given well over a hundred hours per day of engineering time back using the PyTorch deep learning framework powered by NVIDIA DGX-1 computing."

Senior Software Engineer,
Physical Design
NVIDIA

RESULTS

With the trained model, the next step was real-time inference. The team was able to easily set up an inference pipeline, which included NVIDIA V100 GPUs. Without NVIDIA DGX and its container management and resource allocation tools, this would not have been possible.

Every five minutes, lines from log files are sent to the inference pipeline. They no longer need to wait until the job run is complete and the log file is generated, which can sometimes take overnight. Instead, they send the lines to the inference server as they're being added to the log file. Any flagged anomalous lines are fed into the database. The team retrains the model every week for about four hours, so the model can learn from the latest grammar. "We developed a method that requires practically zero user input. Our system can detect as things are changing and will retrain itself as log files with new grammar are fed in," says a senior software engineer at NVIDIA. "We've significantly reduced debugging time and given well over a hundred hours per day of engineering time back using the PyTorch deep learning framework powered by NVIDIA DGX-1 computing."

LOOKING AHEAD

In the future, the team is looking to deliver several feature enhancements to the model, including the ability to classify or group each new error. This way, the engineers only need to debug that type of error once. The team also wants to create a pipeline that lets their users provide solutions to issues. With this in place, when a new error is encountered, the system would automatically search the solution pool to provide relevant suggestions on how to fix the error. And beyond the reasons for failures, the team is implementing ways to analyze the log files for other reasons, such as quality of runs.

"While NVIDIA's physical design team spent about 10 percent of their time debugging prior to using deep learning, today, that's been reduced to 5 percent. We anticipate that this will even further decrease as we evolve our model and more teams look to adopt the same solution," says the NVIDIA engineer. "Industry wide, the impact of using this methodology is huge—on average, software developers spend 35-50 percent of their time validating and debugging software. The cost of debugging, testing, and verification is estimated to account for 50-75 percent of the total budget of software development projects, amounting to more than $100 billion annually." [1]

> "Using AI to automatically recognize valid errors versus false positives amongst an ocean of process log files delivers tremendous savings in time and employee efficiency, while increasing business agility."
>
> Director of Engineering for VLSI
> NVIDIA

"Almost every single company in the world running servers have hosts of applications that generate massive log files. Using AI to automatically recognize valid errors versus false positives amongst an ocean of process log files delivers tremendous savings in time and employee efficiency, while increasing business agility," says NVIDIA's director of engineering for very-large-scale integration (VLSI).

Beyond the design team, the testing team is also looking to adopt the model. GPUs are stress-tested to understand how fast they can run. The test involves making them ray trace various images, applying reflections to reflective surfaces like water, glass, mirrors, and metal. At a certain point when the GPU has been stressed too far, pixelation of the image occurs. Currently, the testing team manually looks at each picture, which is very time consuming. They're looking to apply the same deep learning methodology to train the model to recognize a good versus bad image and understand what each picture should look like based on the pixels surrounding it. This would significantly speed up testing time and remove subjectivity.

www.nvidia.com/dgx

[1] O'Dell, Devon H. "The Debugging Mindset: Understanding the Psychology of Learning Strategies Leads to Effective Problem-Solving Skills." *Association for Computing Machinery*. March 22, 2017.