

# BREAKTHROUGH PRODUCTIVITY WITH MEMORY SNAPSHOTS



Author: Justin Warren

April 2021

# W P

## BREAKTHROUGH PRODUCTIVITY

Moving large datasets into and out of memory presents a substantial challenge for modern workloads including AI/ML, animation, and analytics. Datasets grow ever larger as raw storage costs continue to decrease, but active use of data still requires it to be loaded into memory.

Technologies such as Intel's Optane 3D Xpoint persistent memory provides an affordable way to vastly increase the amount of memory physically available within a server, but this is only half the story. Memverge's memory snapshots bring a tried and trusted mechanism from storage arrays into the big memory era. Memory snapshots open up a range of new workflow possibilities that provide non-linear benefits to organizations that use large datasets every day.

With memory snapshots, moving data to storage becomes a choice rather than a limitation.

### THE TROUBLE WITH TIME

Multiple workload types—such as AI/ML, animation, and genomics—suffer when saving and loading datasets slows down the process pipeline substantially. There are two major contributors to this slowdown: setup time and cycle time.

Saving data from memory to storage in one process stage in order to transfer it to another stage creates a save/load cycle that can be 15 minutes or more. In a lengthy process involving 10 or more stages, this relatively small setup delay can increase the duration of the end-to-end process by multiple hours. For work that needs to run through the entire process frequently, a lengthy setup time can seriously hamper the efforts of entire teams of people.

Similar to capital intensive industries, keeping the expensive 'plant' operating at high utilisation is an important driver of productivity. Large teams of

expensive specialists with rare skills get bored when they spend hours each week just waiting for the infrastructure to keep up with them. They are held back by inferior tools.

The damage done is to more than mere industrial productivity. Creativity and discovery are crushed beneath the oppressive weight of boredom. Boredom borne of waiting for inferior tools to finish tedious, though necessary, setup tasks.

For iterative and exploratory work such as data modelling and animation, pausing to reload data from storage to memory also increases the cycle time for each stage. Creative work involves trying things, making mistakes, noting the effects and trying again. Every additional iteration leads to improvements, and the flexibility afforded by low-risk trials increases our willingness to take creative risks. Risks that can pay off handsomely when we stumble across those happy accidents that lead to a *Eureka!* moment.

We have all felt the frustration of waiting for our computer to perform some menial task like installing updates or completing a reboot when we really want to get something done. Multiply this feeling by all the people in a large animation studio or genomics research facility. Beyond the mere dollar costs, the damage to human motivation and morale can be devastating.

Instead, we can use MemVerge's memory snapshots to all but eliminate these setup tasks and free the humans to concentrate on the interesting work they want to do.



# W P

## FASTER THROUGHPUT WITH FAST SETUP SNAPSHOTS

MemVerge memory snapshots function similarly to the storage-based snapshots we're already familiar with, but using memory instead of storage arrays. Just as storage snapshots can provide a restore point that is faster to use than streaming recovery data from tape or a remote object storage system, memory snapshots provide a recovery point that can be restored much faster than loading data into memory from storage.

For certain workloads, the sheer size of the datasets—terabytes or even petabytes of data—mean loading the data from even very fast flash storage takes many minutes. For example, a generation 3 PCIe interface with 8x lanes could sustain around a 63GB/s transfer rate, but a 12 terabyte dataset would still take over 3 minutes to load into memory, if the flash storage could even supply this bandwidth.

Imagine having to wait 3 minutes every time you need to log into your computer in the morning. Now imagine having to wait 3 minutes every time you need to change application.

For work that needs to move through multiple stages, lengthy setup times slow down the overall process. These setup times also create incentives to pack as much work as possible into overly complex stages in an attempt to avoid the setup costs. This leads to needlessly complex process steps that become difficult to maintain, and the inevitable breakdowns create yet further delays.

For more interactive processes like animation where teams of humans are forced to wait for these setup tasks to complete, the frustration can be palpable. There are only so many coffee breaks you can realistically take before your constant vibrating and non-stop talking annoys everyone around you.

The effect on productivity is more than just the simple accumulation of setup times and the frustrations of dealing with broken systems; the interruptions take you out of an enjoyable flow state, and many people require time to reassemble their mental model of what they were doing and why. The tools themselves become the focus instead of the work they enable.

In-memory snapshots free us from these constraints when dealing with large datasets. Instead of moving data in and out of memory, it stays where it is. An in-memory snapshot provides a rapid restore point, and a space efficient clone makes the dataset available to the next stage in the process.

There is no longer a setup time penalty for decoupling a complex process into multiple stages. Each stage can be optimised to perform a discrete function extremely well before the dataset is moved—still in memory—to the next stage. With clear interface designs, large changes can be made within an individual stage independently of all the other stages, allowing multiple teams to work on the same data simultaneously.

The copy-on-write design of MemVerge's snapshots also provides a space-efficient method of sharing a single large memory space between multiple processes, just as space-efficient storage means more applications can share the same pool of raw storage. A single large dataset can be accessed by multiple read-dominant processes using cloned duplicates that consume no additional raw space.



# W P

## RUNNING MORE EXPERIMENTS

In-memory snapshots provide a transformative experience for individual creative tasks involving large datasets, such as data analysis or computer graphics and animation.

For an animator, experimenting with different ideas to try them out is a central part of the creative process. Ideas need to be tried to see if they work in practice; no matter how great the idea may seem in our head, sometimes when we see it realized it doesn't quite work and we need to try again. Sometimes we might be able to tweak and enhance the idea, and other times we need to simply give up and backtrack to a previous point.

For this kind of interactive experimentation, setup times of even 30 seconds can be enough to induce huge frustration. If the time taken to restart is large, there is a huge temptation to avoid risky experiments that may not work, because the cost of failure is just too high. This psychological barrier can constrain creativity, and the best ideas may never be discovered simply because they were never tried.

For those working at the cutting edge of animation, the animation tools themselves are often bespoke creations that are being actively worked on in parallel to the creative work itself. The tools can be unstable, liable to crash partway through an animation task, and having to wait 15 minutes to restart a painstaking compositing task can be frustrating to say the least. Now imagine having to do this multiple times a day.

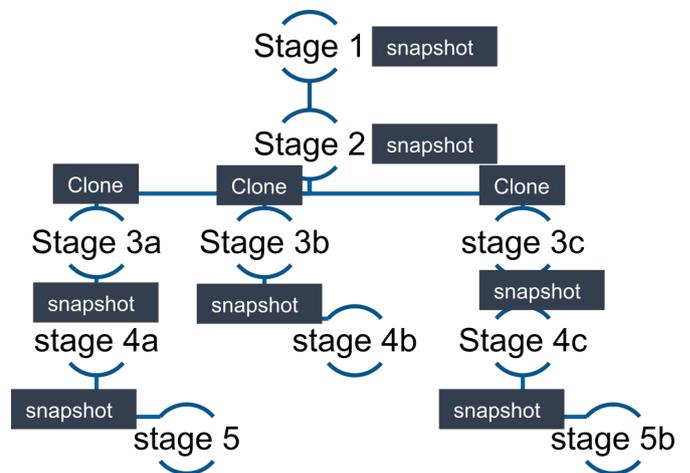
In contrast, imagine having a restart take less than a second before you can continue where you left off. Imagine how many additional experiments you could try when you know the cost of undo is as small as CTRL-Z? What ideas would you be willing to test when freed from the drudgery of waiting 10 minutes for the models to load in each and every time?

The rapid recovery made possible by MemVerge in-memory snapshots provides more than a mere time saving. It provides a qualitative improvement to the human experience of working on something. It removes the boring drudgery from something that is supposed to be fun.

## CASE STUDY: GENOMICS

Single cell RNA (scRNA) sequencing analysis uses a computational model with large datasets, and a multi-stage pipeline with multiple intermediate stages. As the data is analyzed, branching off these intermediate stages is used to perform what-if analyses.

Traditionally, each stage of analysis involves flushing the state of the dataset from memory to storage before



Using snapshots and clones for parallel processing.

re-importing it from storage into memory in the next stage. Each flush and reload cycle can take upwards of 15 minutes to complete, which is a significant proportion of the full process cycle time in a typical 11 stage process. MemVerge memory snapshots reduce this reload time to around 1 second per stage, almost 100 times less idle time.



# W P

Such a reduction in cycle time is transformative. It is a qualitatively different experience for the scientists using the system.

The tools used by data scientists also tend to be highly customised: bespoke programs in Python and R are common. Like all new software, these programs frequently contain bugs. Iterating rapidly to fix bugs in the analysis programs is hampered by long restore times of 15 minutes or more. By using memory snapshots, researchers are able to quickly roll back to a snapshot and re-run their analysis with updated, hopefully bug-free, code. A far greater proportion of researcher time is then spent on active work, not waiting for infrastructure to restore data.

A faster cycle time also enables a greater quantity of parallel experimentation. At each stage, snapshots can be quickly cloned to provide multiple branches for analysis using different parameters, which can proceed in parallel. Each parallel stage enjoys the same rapid cycle time with the added benefit of space-efficiency thanks to the copy-on-write nature of snapshots.

Traditional storage-based, space-efficient snapshots are unsuitable for these workloads due to the need to keep datasets entirely in memory. Flushing the in-memory representation of data to storage in order to take a space-efficient snapshot requires pausing the workload while the flush happens, and while the storage snapshot itself may be quite rapid, the data must then be reloaded back into memory. This process of flushing to storage and reloading means the space efficiency of the storage is then lost, and much greater memory amounts are required compared to in-memory space-efficient snapshots.

## FASTER RECOVERY FOR LOWER RISK

The probable cost of failure is a factor in any

assessment of whether to undertake an activity. The probable cost consists of two components: the likelihood, and the magnitude. Much risk management focuses on reducing the likelihood of failure through risk avoidance but we can also reduce the risk-weighted cost by reducing the magnitude of the per-incident cost.

If I know I can quickly and simply recover from failure, I am far less concerned with avoiding the failure. At the limit, a per-incident cost of zero means it could happen any number of times and we simply do not care about the failure because it has no impact.

If a failure has no impact, can it really be said to be a failure any more?

The total cost reduction is twofold: the direct, per-incident cost reduction, but also the reduction in overhead costs associated with managing risk avoidance.

Many organisations develop complex and unwieldy processes to avoid making costly failures. This investment makes sense where the overhead of such processes is less than the risk-weighted cost of a small number of high-impact failures. However, these overheads can be greatly reduced—if not eliminated entirely—if the cost of failure is negligible.

In fact, an investment in reducing the cost of failure, such as by making recovery quicker and more reliable, is often a better investment. Investments in risk avoidance rarely eliminate risk completely. There is always a chance, no matter how small, that a significant magnitude risk will occur. An unforeseen confluence of circumstances may trigger a failure, a completely new Black Swan event may strike, or you may simply be unlucky one day. If the magnitude of the impact is substantial, all the careful risk avoidance may seem like a waste of time.

Conversely, investing in reducing the impact of failure and making recovery faster will be beneficial regardless of the cause of the failure. Whole classes of causes can be



# W P

thus ignored if they result in the same kind of failure, because recovery from that kind of failure is the same.

Data corruption from a program crash, a programmer bug, or equipment malfunction all trigger the same action: recover from the last memory snapshot. The restoration is fast and predictable, and processing can continue with barely a pause for breath.

## JEVON'S PARADOX

Jevon's Paradox states that when demand is elastic, if the amount of a resource required for any one use decreases, the total use of that resource will increase.<sup>1</sup> Therefore when the cost of individual failures is small, we will be able to fail more often. These kinds of failures are what drives innovation.

Reducing the effective cost of failure opens up a whole new set of possibilities, and the benefits are non-linear. We do not know in advance which experiment we run will result in outsize benefits, but we do know that by running more experiments we are more likely to find an experiment that provides those outsize benefits.

Using MemVerge in-memory snapshots to reduce the cost of failure through fast recovery thus allows for greater experimentation.

This is particularly important in creative, generative fields such as arts and basic research, where high quality ideas have outsize benefits. These high quality ideas are often obvious in hindsight, but difficult or impossible to predict in advance.

In animation, it could be a breakthrough in how a character's face expresses emotion, and that animation sub-system can be replicated to every other character in the feature. The increased artistic impact of the facial emotion expressed is difficult to assess in simple dollar-cost terms, but experienced animators can tell the

difference. The ability to use a more expressive tool allows the animators to explore artistic ideas that were not possible before.

Similarly, in data modelling, the ability to seek out more unusual and creative solutions can lead to a breakthrough in effectiveness or efficiency well beyond what was likely with a more conservative exploration of ideas. Drastic changes to model parameters may lead to new insights that would have been missed if a more pedestrian and less exploratory approach was used.

This is how a reduction in per-experiment cost leads to creation of entire fields of possibility, with benefits that are orders of magnitude higher than simple downside risk avoidance. By reducing downside risk and increasing upside risk, combined with the increase in the total number of experiments conducted, outsize benefits are far more likely to be realized. The payoff is far greater than a simple cost reduction exercise.



<sup>1</sup>Wikipedia 'Jevons paradox'.



# W P

## CONCLUSION

Using techniques like MemVerge's memory snapshots reduces the cycle time for experiments and also lowers the per-incident cost of failures.

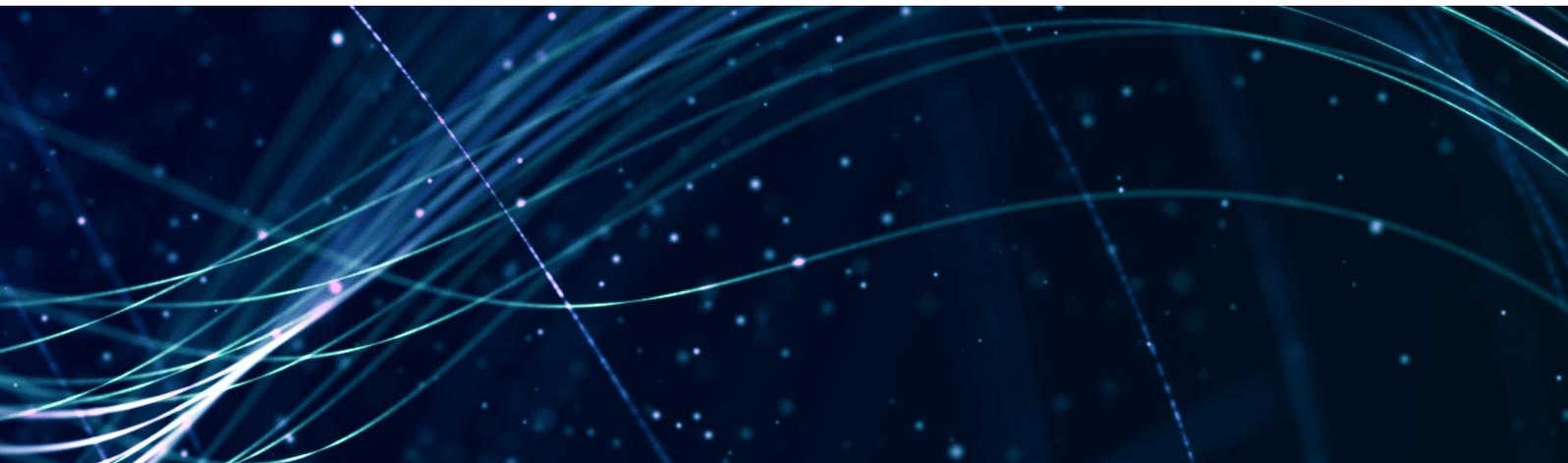
By using memory snapshots to implement straightforward Little's Law optimisations, an organization can achieve substantial increases in throughput. Removing process bottlenecks increases the utilization rate of scarce and expensive resources, such as highly skilled data scientists and animators and the IT infrastructure they depend on.

The benefits of these optimizations are also non-linear. While the direct cost reductions from increased efficiency are easy to calculate, in creative pursuits doing more work in the same amount of time increases the likelihood of stumbling across a breakthrough.

The ability to quickly explore a larger solution space increases the chances of finding a novel solution with substantial value, perhaps orders of magnitude greater than a simplistic efficiency optimisation. Using MemVerge snapshots opens up whole new worlds of potential.



*Using MemVerge snapshots opens up whole new worlds of potential.*



# PN

## ABOUT THE AUTHOR

Justin Warren is Founder and Chief Analyst of PivotNine. He has worked with many well-known companies around the world, including ANZ, Australia Post, IBM, NetApp, Nutanix, Pure Storage, Red Hat, Suncorp, Telstra, and VMware as well as a variety of startups including Atomist, CodeSee, Cloudistics, Datera, Elastifile, Env0, Habrdata, Hasura, Illumio, Isovalent, Manifold, Mattermost, Smallstep, Spectro Cloud, and Solo.io, among others.

Justin has written for a variety of mastheads, including *Forbes.com*, *iTnews*, *CRN Australia* and *The Saturday Paper*.

Justin holds an MBA from Melbourne Business School, and is a graduate member of the Australian Institute of Company Directors.

## ABOUT PIVOTNINE

PivotNine Pty Ltd is a specialist IT consulting firm based in Melbourne, Australia.

PivotNine helps customers to evaluate and select technology products, and to implement effective organisational structures and processes.

PivotNine assists vendors with marketing positioning and messaging, with a focus on data driven marketing methods.

## CONTACT

[enquiries@pivotnine.com](mailto:enquiries@pivotnine.com)

<https://pivotnine.com>





Copyright © 2021 PivotNine Pty Ltd. All Rights Reserved

Reproduction and distribution of this document in any form without prior written permission is prohibited. The information in this document is from sources believed to be reliable. PivotNine disclaims all warranties as to the accuracy, completeness, or adequacy of such information. If any legal issues are discussed in this document, let's be clear that it's not legal advice; PivotNine is not a law firm, and we make no claims that we provide anything even resembling legal advice, so if you want actual legal advice, go and hire a lawyer. PivotNine is not a financial adviser. You should consider seeking independent legal, financial, taxation or other advice to check how information in this document relates to your unique circumstances. PivotNine shall have no liability for errors, omissions, or inadequacies in the information contained in this document. PivotNine is not liable for any loss caused, whether due to negligence or otherwise arising from the use of, or reliance on, the information provided directly or indirectly in this document. Do not tumble dry. This document may, or may not, set fire to your datacentre. The opinions expressed in this document were made at a particular point in time based on information available at that time, and we reserve the right to change our minds as new information comes to hand. If you actually bothered to read this far, send us a note at [ilovelegalse@pivotnine.com](mailto:ilovelegalse@pivotnine.com). PivotNine and the Circular Device are trademarks of PivotNine Pty Ltd. All other trademarks are the property of their respective owners.