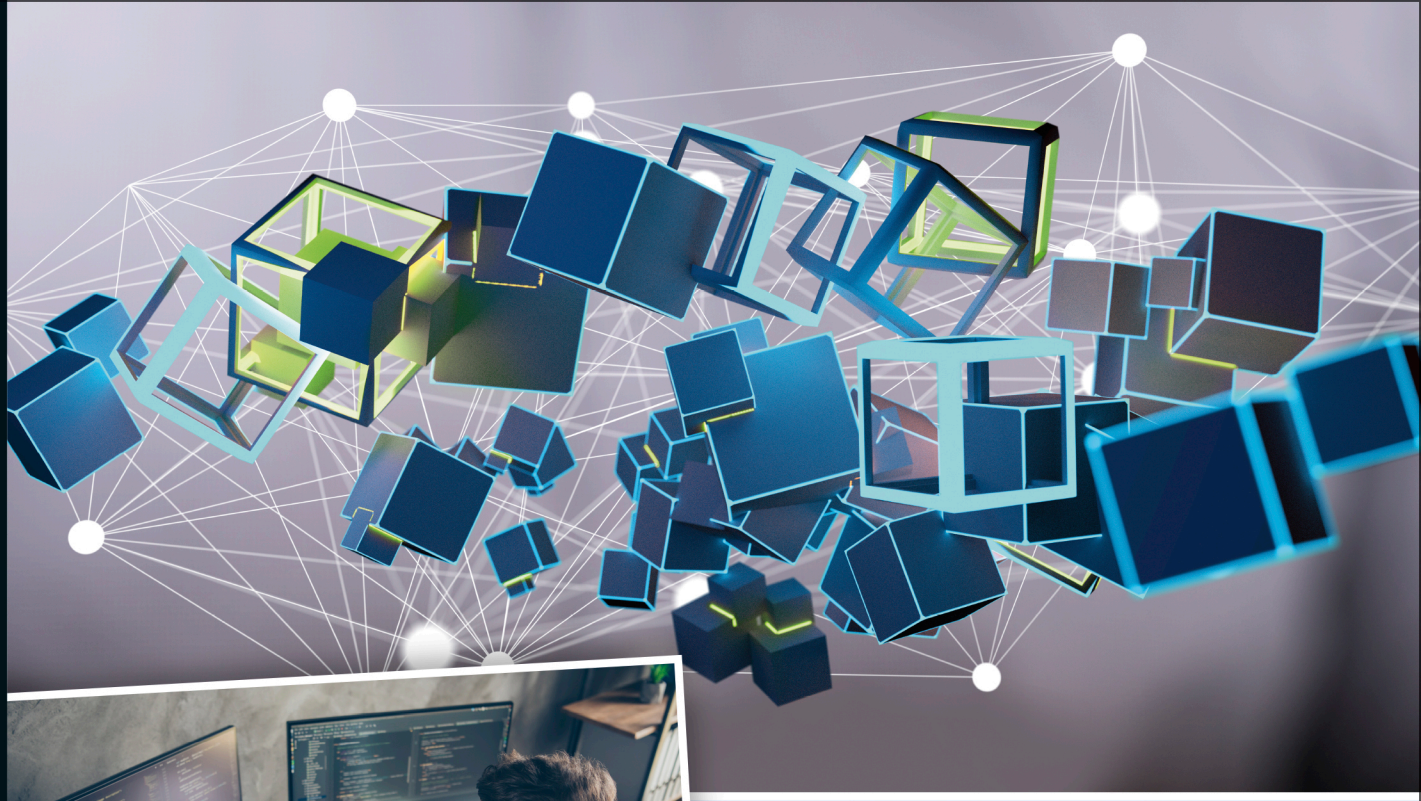
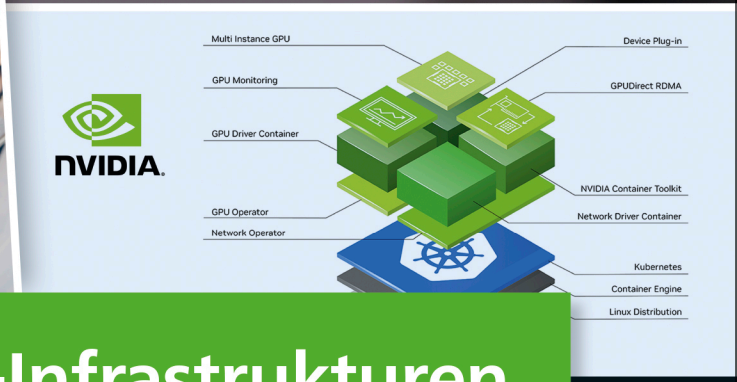




Oliver Liebel

Skalierbare KI/ML-Infrastrukturen

Evaluieren, Automatisieren, Praxis

- Kubernetes- und OpenShift-Cluster mit NVIDIAs Datacenter-GPUs
- Skalierbare und resiliente Infrastrukturen in der Cloud und On-Prem
- Vollautomation und Kosteneffizienz mit IaC und Operatoren

Rheinwerk Computing

Kapitel 1
Vorwort

»Eine echte künstliche Intelligenz wäre intelligent genug, um nicht zu verraten, dass sie wirklich intelligent ist.«
– George Dyson, US-amerikanischer Wissenschafts- und Technikhistoriker

*»Hypes? Similar to as***les. Every day you encounter a new one.«*
– Shaun T., U.S. Special Forces (CAG), ret.

The Long Road – und ein etwas längeres Vorwort

Da wären wir, und es hat lange genug gedauert. Die Arbeit an diesem Buch hat bereits 2019 begonnen, parallel zu meiner 3. Container-Publikation, die Ende 2020 erschien. Und es war ein langer und oftmals und deutlich beschwerlicherer und enervierenderer Weg als die langjährige Arbeit an meinen Container-Themen, die für sich genommen schon einen hochvolatilen Sack sich permanent vermehrender und mutierender Flöhe darstellen, um diese einfache, bildhafte, aber passende Metaphorik zu benutzen.

Anfangs nahm ich noch an, dass sich das Buchprojekt *Skalierbare KI/ML-Infrastrukturen* halbwegs schnell abwickeln lassen würde. Schließlich war das geplante Volumen mit »nur« rund 400 Seiten gerade mal ein Drittel des Umfangs meiner letzten Container-Publikationen. Wie so oft im Leben stellen sich die Dinge in den meisten Fällen dann jedoch eher als komplexer denn einfacher heraus. Das *Warum* der Problematik ist mehreren Punkten geschuldet und zumindest aus einer sehr großen Flughöhe recht schnell erklärt: Ein Kernproblem liegt in der hohen Volatilität skalierbarer Container-Infrastrukturen selbst. Diese stellen jedoch im Unternehmensumfeld die einzig valide Foundation dar, um sowohl reguläre Arbeitslasten wie auch KI/ML-Stacks flexibel und skalierbar vollautomatisch, on demand und für jeden gewünschten Anwendungsfall und Workload provisionieren zu können. Und dieser KI/ML-relevante Stack muss nun noch zusätzlich in zwei Lokationen implementiert werden: im Unterbau (die GPUs in den Servern und deren Management-Systeme in den Hypervisoren) und *on top* im Workload-Layer der Containerisierungs-Plattform.

Klingt so weit erst einmal nicht unüberwindlich. Allerdings war – ergänzend zur ohnehin hohen Volatilität der Containerisierungs-Plattformen – die Entwicklung der GPU-spezifischen Baugruppen des Stacks durch NVIDIA, VMware und Red Hat in den letzten drei Jahren ebenfalls mit einer dermaßen hohen Geschwindigkeit und Volatilität der Komponenten und ihrer Features hinterlegt, dass ich nach Beendigung der Arbeit an einem Themenblock oft

genug den vorherigen, bereits fertiggestellten wieder überarbeiten oder gar komplett neu erstellen musste. Sisyphus lässt grüßen, aber wie üblich sucht sich jeder sein zu tragendes Päckchen meist selber aus.

Wie auch immer – die Dinge im skalierbaren KI/ML-Container-Land sind im zuletzt betrachteten Stand zumindest etwas ruhiger geworden, auch wenn die Entwicklung immer noch rasant ist, und es sind immer noch viele Ecken und Kanten zu meistern.

Der Stand der Dinge folgt nun.

Superlative, Geschwindigkeit, Hypes und die Realität

2022 – ein Jahr der Superlative. Alles ist groß. Aber zum Teil auch wieder klein. Größenwahnsinnige und zugleich kleingeistige Autokraten. Großflächige Corona-Durchseuchung mit hoffentlich kleinem Impact. Große Rezessions- und Inflationsängste, die sich hoffentlich nur zu einem kleinen Teil bewahrheiten. Große Nachfrage nach Gütern und Rohstoffen jedweder Art und ein mittlerweile oft viel zu kleines Angebot.

Aber nimmt man die Aussagen in allen Medien und den üblichen Politiker- und Consulter-Buzzword-Bullshit-Bingo-Runden abseits dessen zusammen, gibt es anscheinend im technischen Bereich nach wie vor nichts Größeres und Wichtigeres als Di-, Meta- oder Omniverses und Machine-Learning-/KI-Systeme oder allgemeiner:

Künstliche Intelligenz.

Nun, da zumindest in der Politik selten echte anzutreffen ist, könnte das theoretisch ein Hoffnungsschimmer sein.

Aber was ist mit den Unternehmen, die ihre Applikationsstacks immer noch nicht auf Basis von Big Data, KI und maschinellem Lernen betreiben? Denn die haben ja – zumindest gemäß dem schon viel zu lange anhaltenden KI-Hype- und Buzzword-Tsunami – mittel- bis langfristig ganz sicher keine Überlebenschance ohne hyper-agil entwickelte, self-optimizing and -repairing KI-Systeme. Die jeden Tag, jede Stunde aktualisiert oder am besten mehrmals pro Woche gleich ganz neu erfunden werden. Rückwärtskompatibilität ist ja auch sowas von gestern, oldschool und voll out. So wie alles, was noch on-prem und nicht in der Cloud läuft.

Ach ja? Ist das so?

Nun ja.

Im Grunde genommen spiegelt der ganze Faster-, Faster-, Faster-Nonsens im IT-Bereich ein Paradigma wider, in das sich unsere gesamte Gesellschaftskultur – leider, muss man sagen – unvermeidlich bewegt. Häppchen müssen nur noch mundgerecht serviert werden, egal ob's hier und da mal nicht so schmeckt oder passt, wie es soll. Hauptsache schnell, schnell, schnell. Trauriges Synonym einer fast schon demagogischen Lebens- und IT-Entwicklung, in der – zumindest aktuell – alles, was wir aufnehmen, so konzipiert und aufbereitet werden muss, dass auch Rezipienten mit einer maximalen Aufmerksamkeitsspanne von dreikommafünf Sekunden nicht sofort gelangweilt sind.

Viele böse Zungen behaupten ohnehin schon länger, und politisch völlig unkorrekt, dass »Agilität« in der IT wahrscheinlich doch nur eine freundliche Umschreibung sei, um Personen mit ADHS-Tendenzen in IT-Projekten unterzubringen. Wie üblich hängt das vom Standpunkt des Betrachters ab. Und Hypes sind erfahrungsgemäß in der Realität irgendwo zwischen dem verortet, was sie und ihre Fanboys versprechen, und dem, was eine (von ihren Verweigerern gern) überkritische Darstellung ihrer Mankos ist.

Viele Probleme – (k)eine echte Lösung?

Und wie üblich wird der zunehmende, großflächige und damit irgendwann relativ günstig verfügbare Einsatz einer innovativen und extrem leistungsfähigen Technik wie KI/ML auch abseits effizienter, legitimer und moralisch korrekter Anwendungen jede Menge Schattenseiten nach sich ziehen:

Denn solange wir auf die gleichen Technologien setzen, werden Leistungshunger und Abwärme von RZs mit KI-tauglichen GPU- und TPU- Systemen in jeder neuen Generation exponentiell durch die Decke gehen.

Aber hey, was soll's, werden sich (zu) viele grenzdebile Individuen mit viel Langeweile und/oder zu wenigen Hirnwindungen sagen: *Green IT ist voll gestern, Klimawandel Fake News und Greta nervt eh nur. Oder?* Und so kann man schließlich auf Knopfdruck und völlig on demand in High-End-Clouds, die minütlich Abwärme in der Größenordnung von New York verbraten, seine Nachbarn und/oder den Rest der Welt mit per Deepfake generierten, personalisierten Bildern und Videos verwerflichen Inhaltes beglücken, die auch gegen Whatever-Coin-Zahlungen ganz sicher nie wieder gelöscht werden.

Schalten wir den Sarkasmus mal wieder ab, aber Sie wissen so gut wie ich, dass dies erfahrungsgemäß nicht so weit abseits der Realität liegt, wie man denken möge. Und wir reden an dieser Stelle noch nicht einmal von vollautonomen Waffensystemen.

Dass die vollautomatische Bearbeitung von Prozessen auf Basis komplexer Daten nicht nur Vorteile bieten kann, sollte auf der Hand liegen. Allein schon die persönlichen Daten eines jeden Menschen bieten ein gewaltiges Potential, sowohl in positiver wie negativer Hinsicht: Stammdaten einer Person aus ihrer bisherigen Krankenakte, Fehlzeiten im Job, Bestellung (nicht)alkoholischer und/oder extrem zuckerhaltiger Getränkelieferungen per Internet, Postings der Person in sozialen Netzen z. B. über (extrem)sportliche Aktivitäten, erlittene physische oder psychische Krankheiten/Traumata oder (un)populäre Gesinnungen, dazu Gesundheits-Apps auf dem Handy oder der Smartwatch, welche die Telemetriedaten des letzten Joggings (oder Körperfunktionen permanent) übermitteln, die vielleicht nicht dem optimalen Vital-Standard entsprechen, und vieles andere mehr. Dies alles eignet sich »hervorragend« dazu, bestimmte Personen oder Risikogruppen beispielsweise automatisch in bestimmte Versicherungstarife einzuordnen, diesen Personen Kredit zu gewähren (oder nicht), sie bei Bewerbungen auszuschließen (oder nicht), bei Beförderungen zu berücksichtigen (oder nicht) und so weiter. Das Prinzip dürfte offenkundig sein.

Oder nehmen wir ein weiteres, mögliches Problemfeld, bei dem die Auswirkungen weitaus drastischer sein können – die vernetzte KI, wie sie bereits aktuell und zukünftig zunehmend auch herstellerübergreifend in KI-gesteuerten Fahrzeugen zum Einsatz kommt. Nehmen wir eine Situation an, in der zwei KI-gesteuerte Fahrzeuge mit hoher Geschwindigkeit unvermeidbar aufeinanderprallen werden, sofern keine Entscheidung der Autopiloten getroffen wird. Situationsbedingt kann nur ein Fahrzeug bzw. Insasse gerettet werden, die Person in dem anderen Fahrzeug muss mit hoher Wahrscheinlichkeit sterben.

In Fahrzeug 1 befindet sich eine 75-jährige Person, in Fahrzeug 2 ein Kind von 7 Jahren, das zur Schule gebracht wird. Wie entscheidet nun die vernetzte KI? Und exakt an dieser Stelle des Gedankenspiels kommen etliche Faktoren hinzu, die erst beim zweiten Blick auf das Szenario klar werden: Kennt das vernetzte KI-System die Krankenakte des Kindes und der älteren Person und weiß bereits, dass das Kind an einer schweren Krankheit leidet, mit einem garantiert tödlichen Verlauf innerhalb eines Jahres, während die ältere Person noch kerngesund ist?

Hat das KI-System gegebenenfalls Kenntnis darüber, dass die ältere Person eine wichtige Person des öffentlichen Lebens oder der Politik ist oder gar zum Vorstand genau des Konzerns gehört, von dem das Fahrzeug und/oder die KI-Systeme produziert wurden? Und wurde exakt für dieses Worst-Case-Szenario bereits eine versteckte Exception bzw. Priorisierung in die Entscheidungsalgorithmen eingebracht? Und was passiert, wenn der Fall »klassisch« andersherum liegt: Die alte Person ist krank, hat keine hohe Lebenserwartung mehr, die Rentenkassen sind leer, die Kosten der Krankenkasse für die ärztliche Versorgung der alten Person sind immens hoch. Das Kind jedoch gehört zu einer wohlhabenden, einflussreichen Familie und ist kerngesund.

Nun, gemäß einem nicht defekten moralischen Kompass sollte jedes Leben gleich viel wert sein. So weit die Theorie, aber allein das gerade gezeigte, kleine Gedankenspiel sollte verdeutlichen, dass die Dinge sehr schnell viel komplizierter werden können. Paart man diese Betrachtungen mit den – seit Anbeginn der Menschheit ständig und erfahrungsgemäß leider immer noch viel zu häufig anzutreffenden – Charakterzügen wie Egoismus und Gier auf jeder Hierarchieebene, wird viel schneller ein Schuh daraus, als man denken mag.

Und dabei reden wir noch nicht einmal von potentiellen Schwachstellen, durch welche z. B. gehackte, autonome Schwerlaster als Terrorwaffen missbraucht werden könnten. Oder von militärischen Drohnen der nächsten Generation mit KI-gestützten Waffensystemen, die zukünftig innerhalb bestimmter Parameter auch ohne die finale Human-Authorization in Sekundenbruchteilen Entscheidungen treffen können. Sie tun dies auf Basis der ihnen vorliegenden Daten sowie der Daten, mit denen diese Systeme trainiert wurden, und müssen dann im ungünstigsten Fall im Millisekunden-Bereich eine Entscheidung über Nutzen vs. Kollateralschaden treffen.

Und wer denkt, dass diese kleinen Gedankenspiele fernab der Realität liegen – dem ist nicht so.

Wir ahnen bereits, wie komplex das Thema abseits der üblichen und oft leider nur rein technischen Betrachtung tatsächlich zu sehen ist.

Foundation – und »mehr Power«

Fakt ist: Die Verarbeitung der Daten, sowohl für die Trainings von KI-Systemen als auch im späteren Einsatz, ist unstrittig hochkomplex. Aber auch wenn alle Data Scientists und Data Engineers, Programmierer und auch alle anderen, die sich um die Vorbereitung und KI/ML-Modellerstellung kümmern, höchst sorgfältig arbeiten und alle Daten maximal transparent und zudem in allen relevanten Belangen korrekt sind, kommt noch ein wichtiger, entscheidender Faktor hinzu.

Die Systeme, die die Daten prozessieren. Leistungsfähige und vollautomatisch skalierbare KI-Infrastrukturen. Hocheffiziente High-Performance-GPUs in Container-Clustern, die dynamisch mit DPUs und CPUs in Software-Defined Datacenters zu einer logischen Super-Compute-Unit verschmelzen, um allen involvierten Teams des Unternehmens das Leben leichter zu machen und nicht komplizierter. Und dem Kunden die Informationen liefern, die er bestellt hat bzw. benötigt – und zwar möglichst ohne einen Prediction-Error, der im ungünstigsten Fall vielleicht Einfluss auf Leib und Leben haben könnte.

Wie es der KI-Forscher Richard Sutton in der Einleitung seines Artikels *The Bitter Lesson* 2019 trefflich formulierte:

> *»Die wichtigste Lektion, die man aus 70 Jahren KI-Forschung ziehen kann, ist, dass allgemeine Methoden, die schlichtweg Rechenleistung nutzen, letztlich die effektivsten sind, und zwar mit großem Abstand. [...] Auf der Suche nach einer Verbesserung, die kurzfristig einen Unterschied macht, versuchen Forscher, ihr menschliches Fachbereichs-Wissen zu nutzen, aber das Einzige, was auf lange Sicht zählt, ist massive Rechenleistung.«*

Oder kurz: *Viel hilft viel*, zumindest im Moment und zumindest, was die Rechenleistung von KI-Systemen angeht. Und damit sind wir bei der zwingend erforderlichen Skalierbarkeit dieser Systeme.

Und genau diese Skalierbarkeit von containerisiert arbeitenden KI/ML-Infrastrukturen war mein Trigger, dieses Buch zu schreiben, da es sehr eng mit meinem primären (und seit fast 30 Jahren Operations-lastigen) Schaffensfeld verheiratet ist: hochskalierbare Container-Cluster-Infrastrukturen.

Dabei ging es mir nicht darum, ein weiteres Fachbuch der Kategorie *Prima, ich habe auf GCP/AWS/AKS/EKS/GKE/Whatever einen Python-Code-Snippet in mein containerisiertes Jupyter-Notebook mit TensorFlow gepastet und es tut irgendwas ...* beizusteuern, zusätzlich zu den gefühlten Millionen Exemplaren, die auf dieser Welle bereits mitschwimmen.

Hypes und die eher selten auseinanderbrechende Realität

Mir ging es vor allem darum, als jemand, der seit fast drei Jahrzehnten sehr tief in hochkomplexen Themengebieten der IT verwurzelt ist, zu analysieren, was wirklich hinter dem ganzen Hype steckt. Kritische und unbequeme Fragen abseits von oftmals blindem Opportunismus und maximal enervierend gebetsmühlenartig repetierten »Mit KI/ML lösen wir alle Probleme«-Mantras zu stellen.

Denn das Konzept, einen KI/ML-Insel-Stack von einer KI-Workstation eines Data Scientists *mal eben* in eine auto-skalierbare, produktivtaugliche und Multi-Tenant-fähige Cluster-Umgebung zu verpflanzen, erfordert mehr als eine Hollywood-geprägte Illusion von KI-Systemen und ein paar knallige, cloudaffine Buzzwords beim dritten Cappuccino im legeren Entscheider-Meeting.

Was wird auf der Infrastruktur-Seite wirklich benötigt, um KI/ML-Systeme flexibel, effizient, stabil und sicher zu betreiben? Und zu welchem Preis? Es geht darum, zu hinterfragen und zu analysieren, *was* implementierungs- (ausdrücklich nicht coding-) und infrastrukturtechnisch *wie* für *wen* »geht« – und was nicht. Und vor allem: mit welchem *Automationsgrad*. Denn das ist der wichtigste Schlüssel für performante, skalierbare, containerisierte KI/ML-Cluster. Und zu schauen, wo die Reise hinführen kann. Mit allen positiven und leider auch nicht wenigen negativen Aspekten.

Es geht um die explizite Beleuchtung von oftmals nicht unerheblichen Implementierungsproblemen, insbesondere für Unternehmen, die sensible Daten ihrer KI/ML-Workflows in der nach wie vor hochbedenklichen Sicherheit von Everything-happy-Everything-a-a-S-ML-Clouds prozessieren möchten – und im Cloud-Anwendungsfall noch dazu permanent Kosten auftürmen, die jenseits von Gut und Böse liegen. Aber auch im Self-Hosted-KI/ML-Bereich ist leider nicht mehr viel los mit eitel Sonnenschein, sondern in vielen Teilen eher der permanente Link nach */dev/null* für Dollars, Euros und Effizienz angesagt – wenn sorgfältige Planung fehlt oder einmal mehr auf Entscheider-Ebene dank hart antrainierter Beratungsresistenz gekonnt ignoriert wird.

Es geht darum, wie hoch der Grad der *Vollautomation* für containerisierte, skalierbare KI/ML-Cluster im betrachteten Stand wirklich ist bzw. sein kann. Denn nur das zählt in ernstzunehmenden Unternehmensimplementierungen. KI-Systeme kann seit einigen Monaten oder Jahren plötzlich jeder implementieren, wenn man dem Geschwurbel der Medien und üblichen Verdächtigen lauscht. Wie üblich liegen zwischen Anspruch und Wahrheit in der Regel oft Welten. Greift man tiefer, besitzen nur die wenigsten das Know-how um die echte, hohe Kunst: die Implementierung massivst skalierbarer Container-Cluster für KI/ML-Applikationen mit maximaler Vollautomatisierung auf jedem Level, von der Infrastruktur bis zum operatorgestützten Rollout der (v)GPU-Ressourcen und ML-Stacks, egal ob on-premises oder in der Cloud.

Es geht um Fragen und Antworten zu Konzepten, ROI und LTS. Mit einem konsequenten Fokus auf potentielle Strategien für ML-Infrastrukturen in Unternehmen. Mit Betrachtungen und Analysen zu Auswirkungen auf das Unternehmen durch eine Umstrukturierung auf containergestützte KI/ML-Systeme. Ob sich der Einsatz von Machine-Learning-Systemen für das eigene Unternehmen rentiert. Und wenn ja, in welchem Umfang.

Und es geht darum, dass auch diejenigen von uns, die Python nicht ad hoc mit vierhundert Anschlägen pro Minute coden können, und auch die, die bei Begriffen wie TensorFlow, Keras, PyTorch und Ähnlichem nicht sofort Speichelabsonderungen produzieren, eine realistische

Einschätzung darüber erhalten, was KI/ML-Infrastruktur-technisch im betrachteten Stand im Unternehmensumfeld umgesetzt werden kann. Und was in Grimms Märchenstunde bzw. -Tonne bzw. den großen, ewigen Schwurbel-Bullshit-Leitfaden aller Sales-Fraktionen gehört.

Faster, faster … und kein Ende. Leider auch für die Kosten.

Denn insbesondere in der immer schnelllebigeren Container- und KI/ML-Welt haben Informationen rund um Hardware, Software, Verfahren und Konzepte leider eine immer kürzere Halbwertzeit.

Eine höchst bedauerliche Problematik, auf die ich bereits in meinen letzten vier Container-Publikationen mehr als ausdrücklich hingewiesen hatte und an der sich bis heute herzlich wenig geändert hat. Im Gegenteil. Aber was soll's – denn den Sales-Fraktionen von NVIDIA, AWS, GCP, Red Hat, VMware und Co. bereitet das meist keine mächtigen Kopfschmerzen. Allen übrigen Beteiligten sehr wohl. Denn im Grunde gilt nur noch eine Regel: dass nichts mehr gilt. Oder in Langform: Das meiste, was gestern noch superhip und State of the Art war, ist heute schon oft genug nur noch alter Krempel, der keinen mehr wirklich interessiert.

Und das trifft auf fast alle Beteiligten zu – mit Ausnahme des Endkunden. Denn der würde sich gern mal eine etwas langsamere Pace wünschen, Innovation hin oder her.

Fatalerweise müssen sich die Unternehmen aber ebendieser hochvolatilen Gemengelage permanent neu stellen und anpassen, und dies oft zu einem – im wahrsten Wortsinne – hohen Preis.

Kein anderes IT-Geschäftsfeld ist im betrachteten Stand mit derartig hohen Kosten hinterlegt. Auf der anderen Seite: Die relevanten Unternehmen, die den Zug verpassen, werden sich irgendwann keine Sorgen mehr um Kosten machen müssen bzw. können. Im Grunde heißt es wie üblich »am Ball bleiben« oder wenigstens »endlich einsteigen« – aber es wird von Jahr zu Jahr erfahrungsgemäß eher schwieriger denn einfacher:

Kleinere Unternehmen machen sich meist Gedanken, wie die eigene Infrastruktur mit möglichst geringem finanziellem Aufwand KI/ML-tauglich gemacht werden kann, z. B. durch Aufrüstung einiger On-Prem-Systeme mit zusätzlichen GPUs und verstärkter Kühlung und ergänzend den einen oder anderen lastintensiven Trainings-Workload in der Cloud, stecken dann aber erfahrungsgemäß leider allzu oft in einer teuren Pseudo-Kreativ-Sackgasse voller loser Enden fest.

Etliche Mittelständler pendeln zum Teil nach wie vor in fantastisch-kreativem Herumgeeiere eher unentschlossen zwischen Cloud und On-Prem, Letzteres mit dedizierten, gekauften oder gemieteten GPU-Servern oder aufgebohrten Bestandsservern in den Unternehmens-RZ, ohne wirklich nach vorne zu kommen.

Und im Konzernbereich sieht die Sachlage auf den ersten Blick oft klarer aus, aber seien Sie versichert – das ist sie meist nicht. Viele sind überstürzt in die scheinbare und trügerische

Sicherheit »*KI/ML? Alles kein Problem. Vertrauen Sie uns …*« verschiedener Cloud-Provider (man will ja schön diversifizieren) abgerauscht und kämpfen nun mit providerspezifischen Implementierungsproblemen, mangelnder Konfigurierbarkeit der GPU-Nodes, den allgegenwärtigen cloudtypischen und -bedingten Sicherheitsproblemen und nicht zuletzt sehr hohen Kosten.

Und wieder die unbequeme Realität, aber hübsch aufbereitet

Womit wir wieder bei der allgegenwärtigen Cloud und ihren Anbietern wären. Und selbst die müssen sich mehr und mehr der harten Realität stellen, welche uns seit vielen Monden und in echter 24/7-HA nonstop Pandemien, Kriege, Krisen, Rohstoff- und Energieknappheit und einiges mehr zur täglichen Erbauung beschert. Aber die Provider verbiegen die Realität wie üblich mit einem (jedoch nur auf den ersten Blick) gekonnten Griff in die Werbe-Trickkiste.

Microsoft zog Mitte 2022 als Erster die Handbremse und kündigte ab sofort eine längere Nutzungsdauer für Cloud-Hardware an. Konkret verlängern die Redmonder ab 2023 die Gesamtbetriebsdauer ihrer CPU- und GPU-Server bis zum Austausch gegen neuere Systeme von vier auf sechs Jahre. Das maximale sinnfreie Werbe-Blabla, dass für den nun verlängerten Betriebszeitraum »*Investitionen in unsere Software [erfolgen], die den Betrieb unserer Server- und Netzwerkausrüstung effizienter machen*« (Zitat Microsoft, untermalt von brüllendem Gelächter gestandener RZ-Admins weltweit), gehört sicher in jede Vorstandssitzung, aber technisch betrachtet auch nur nach */dev/null*.

Aber was soll's: Nicht zuletzt die Shareholder wird es freuen, da Microsoft dank der verlängerten Abschreibungsdauer allein im Fiskaljahr 2023 Einsparungen von fast 4 Milliarden erwartet. Auch Google (Verlängerung von drei auf vier Jahre) und AWS (Verlängerung von vier auf fünf Jahre) haben längst ähnliche Schritte angekündigt.

Neben allen Finanztricks, um besser durch die Krise zu kommen, bedeutet das jedoch auch unter dem Strich, dass Unternehmen, die ihre KI/ML-Strecken ganz oder in Teilen in der Cloud betreiben, schlichtweg länger auf CPUs und GPUs neuester Bauart verzichten müssen und damit auf eine gegebenenfalls deutlich bessere Performance und/oder Energieeffizienz.

Ja, stimmt: Abstrakt betrachtet, klingt das alles zunächst sehr nach einer Diskussion Pest vs. Cholera. Ich könnte jetzt sagen: *Alles gut, Freunde – keine Sorge, dem ist nicht so*.

Aber das würde leider nicht der Realität entsprechen.

Denn es geht im Grunde eigentlich und letztlich nur noch darum, wie man die IT des eigenen Unternehmens (sofern KI/ML-Szenarien dort von hoher wirtschaftlicher Relevanz sind, was auf immer mehr Unternehmensbereiche zutrifft) in diesen technisch, betriebs- und finanzwirtschaftlich hypervolatilen Zeiten strategisch so aufstellt, dass der geringstmögliche wirtschaftliche Schaden entsteht.

Denn seien Sie auch bezogen auf diesen Punkt versichert: Wenn Sie in der Welt der KI- und ML-Cluster-Infrastrukturen ein oder zweimal design- und entscheidungstechnisch falsch abgebogen sind, kann das bereits die sprichwörtliche Road to Financial Ruin sein.

Falls Sie meinen, dies wäre eher eine Übertreibung – willkommen im realen Leben. Es gab in den vergangenen Jahren genügend Unternehmen, die diese leidvolle und teure Erfahrung machen mussten.

In eigener Sache

Und damit die Leserinnen und Leser, die mich noch nicht aus meinen acht bisherigen Publikationen kennen, wissen, von wem diese Aussagen kommen: Ich bin jemand, den andere vielleicht als Spezialisten im RZ-/Cloud-/Großkunden-Bereich für High-Availability-Cluster, Software-Defined Storage, Verzeichnisdienste sowie für hochskalierbare, vollautomatisierte Container-Cluster und GPU-beschleunigte Microservice-Infrastrukturen von großen Unternehmen und international operierenden Konzernen bezeichnen würden. Als Systemarchitekten und oft genug auch als Problem-Fixer für bestimmte Bereiche der IT von Unternehmen und Konzernen.

Aber im Grunde bin ich unter dem Strich nichts anderes als ein IT-Veteran, der unzählige Hypes hat kommen und gehen sehen. Und genau daher geht es mir, wie in all meinen bisherigen Publikationen, vor allem darum, eine möglichst realistische Einschätzung abzuliefern. Darüber, wo wir implementierungstechnisch im Bereich der KI/ML-Infrastrukturen wirklich stehen, ob und wie für das eigene Unternehmen ein gangbarer Weg durch den KI-Infra-Dschungel gefunden werden kann und für wen sich der Einsatz dieser Systeme lohnen kann.

Aber selbst im Vergleich zu meinen vier bisherigen Publikationen rund um das ebenfalls hochvolatile Themengebiet Container-Cluster kann jede Publikation, die sich im betrachteten Stand seriös mit der Thematik skalierbarer KI/ML-Cluster-Infrastrukturen beschäftigt, trotz aller Genauigkeit immer nur eines sein – eine Momentaufnahme. Und genau deswegen habe ich dieses Mal einen etwas anderen Ansatz gewählt, nämlich einen, der etwas weniger mit praktischen Beispielen und etwas mehr mit Theorie und vor allem Strategie hinterlegt ist.

Denn wie ich im Laufe der zähen und zum Teil leidvollen Recherchen und Tests (danke an dieser Stelle noch einmal an die vielen netten Menschen von NVIDIA, insbesondere Erik Bohnhorst, die mir dabei fast drei Jahre lang zur Seite standen) für dieses Buch erkennen musste, sind zwar praktische Betrachtungen, also das *Doing*, absolut essentiell und unabdingbar, um bestimmte Sachverhalte erstmalig zu verinnerlichen.

Jedoch spielt in diesem Fall der pure Technik-Kontext, d. h. die Konfigurationen bis hin zum letzten kleinen Bit, strategisch eine etwas untergeordnete Rolle. Weil er bzw. es sich unglaublich schnell verändert – noch schneller als in »reinen« Container-Clustern, die für sich genommen schon volatil genug sind und hier lediglich die Foundation für die KI/ML-Stacks darstellen.

Und betrachtet man die Entwicklung im RZ- und Cloud-Bereich einmal ohne alte Muster und mühsam etablierte Scheuklappen, so ist der gute alte Server doch schon lange nicht mehr die Compute-Unit: Es ist längst das bereits erwähnte, vollautomatisierte *Software-Defined Datacenter*, mit jeder Menge CPUs, GPUs, TPUs und DPUs.

Was im Bereich von skalierbaren, containerisierten KI/ML-Infrastrukturen wirklich zählt, ist insbesondere das Verständnis, wie welche Komponenten in welchem speziellen Szenario arbeiten und wie sie ineinandergreifen. Ist das verinnerlicht, ist das Verständnis für die Executive-Komponenten von Hard- (GPUs/TPUs) und Software (Container-Cluster und Operatoren) relativ schnell auf den nächsten Evolutions-Level adaptiert. Und damit portierbar.

Ja, geschenkt, Freunde – pünktlich zum Release von NVIDIAs erster Quanten-GPU mit Ultra-Freon-Kühlung gibt's dann eine leicht aktualisierte Auflage.

Wie auch immer, gehen wir's an. Denn die Arbeit erledigt sich leider noch nicht von selbst.

Aber genau daran arbeiten wir.

Und wer das Motto lieber im Buzzword-Sprech für die Vorstandsrunde hätte:

Hyperautomation mit KI.

Ist ganz sicher bis zum Ende dieser Woche noch superhip und voll angesagt.

Danach, naja …

Ah, der Cappuccino ist fertig.

Danksagungen und Widmung

Mein besonderer Dank geht an viele Mitarbeiter von NVIDIA aus dem Datacenter-GPU-Bereich weltweit, die mich in den letzten Jahren tatkräftig unterstützt haben, insbesondere: Erik Bohnhorst, Christopher Desiniotis, Milan Diebel, Francis Guillier, Shiva Krishna Merla, Rajeshka Rao, Thomas Remmlinger, Thomas Wernicke und viele andere mehr, die hier nicht explizit benannt sind. Ebenso gilt mein Dank dem Red Hat EMEA Partner Team.

Für T.Y. – in tiefstem Dank für Deine zeitlosen, unerschütterlichen Weisheiten.

1.1 Vorbemerkungen

Im Folgenden finden sich vermehrt Anglizismen (welche in der Regel auch erklärt werden), die sich im Rahmen aktueller und fachspezifisch fortgeschrittener Applikationen/Publikationen ohne kilometerlange und oft unpassende deutsche Um- und Beschreibungen oft nicht umgehen lassen.

Kapitel 4
NVIDIA-Datacenter-GPUs und mehr – technischer Background

»9.7 TFLOPS for FP64? Sounds great, but, eh – can it run DOOM?«

– Typischer Foren-Kommentar bei so ziemlich jeder Ankündigung einer neuen High-End-GPU-Karte

In diesem Abschnitt befassen wir uns konkreter mit der Hardware, sprich – den NVIDIA-Datacenter-tauglichen GPUs, den jeweils dafür geeigneten Bereitstellungsverfahren, ihrer Performance und etlichem anderen mehr. Mit dem Bereitstellungsverfahren ist im einfachsten Fall beispielsweise das dedizierte Durchreichen einer GPU an eine VM gemeint oder softwarebasierte GPU-Partitionierungen per Virtual GPU (vGPU), hardwarebasierte Partitionierung (MIG) mit einer besseren Tenancy, GPU-Sharing oder Kopplungen multipler GPUs, lokal oder über das Netzwerk, oder auch verschiedene Kombinationen der vorgenannten Verfahren, um unterschiedlichste Anwendungsfälle abdecken zu können.

4.1 NVIDIA und ML-Cluster

Nun, wo fängt man an? Vielleicht mit einer Frage:

Lieber NVIDIA-Aktien anstelle von Goldbarren? Nein, das soll durchaus kein polemischer Kommentar zu Anlagestrategien sein, sondern liegt näher an der Realität, als Sie vielleicht denken mögen, wie im Folgenden erklärt werden wird. Denn kaum ein Unternehmen hat sich strategisch in den letzten Jahren in eine ähnliche monopolartige Position gebracht, wenn es um KI/ML-Belange im RZ-Bereich geht, wie NVIDIA.

NVIDIAs GPU-Accelerators wie die neue Hopper-Generation oder ihre Vorläufer wie die Ampere-Generation oder deren Tesla-Vorgänger (Turing-Generation) sind aus Rechenzentren kaum noch wegzudenken. Datacenter-taugliche GPUs werden – je nach GPU-Modell – zwar auch noch in bestimmten Unternehmen für Remote-Desktop-Szenarien, also »echte« grafische Workloads eingesetzt, in zunehmender Anzahl jedoch primär nur noch für *Accelerated Computing*, also KI/ML-Einsätze mit GPU-Beschleunigung.

Dabei existieren verschiedene Möglichkeiten, die Rechenleistung der GPU zu nutzen. Während für reine ML-Entwicklungs-Workstations beispielsweise dedizierte High-End-Grafik-

karten wie eine High-End-Karte der RTX-Serie per *Passthrough* angebunden werden können, bleibt das Manko, dass die jeweilige GPU nur von *einem* Nutzer bzw. exakt einer realen oder virtuellen Maschine für ML-Workloads verwendet werden kann. Für geclusterte und skalierbare Szenarien in Unternehmen wird also eine andere Herangehensweise benötigt.

Insbesondere Kubernetes-/OpenShift-basierte Container-Cluster mit Pod- und Cluster-Autoscalern sind daher eigentlich prädestiniert, um bei entsprechender Lastanforderung automatisch in die Breite zu skalieren, um so für alle (KI/ML-)Anwendungsfälle genügend Rechenkapazitäten bereitstellen zu können. So können z. B. multiple GPU-fähige Anwendungen automatisch horizontal skaliert werden, um große Datenlasten parallel prozessieren zu können – natürlich vorausgesetzt, dass die Anwendung bzw. Architektur dies unterstützt.

Logischerweise gibt es auch alternative Verfahren wie das bereits erwähnte, dedizierte Passthrough (1:1-Bindung zwischen VM und GPU-Accelerator), die zwar technisch ebenfalls greifen, jedoch in Clustern (siehe das Thema »Skalierung«) für Accelerated Computing selbst mit Verfahren wie VMwares *Dynamic DirectPath IO* (siehe Abschnitt 4.5.3) nicht wirklich effizient sind. Es werden daher *shareable* GPU-Ressourcen benötigt, und hier startet die Geldruckmaschine für Cloud-Betreiber, Virtualisierer – und nicht zuletzt NVIDIA.

Betrachten wir das »Warum«.

Cloud

Nahezu alle Cloud-Provider lassen sich mittlerweile ihre VM-Instanztypen mit durchgeschleifter GPU/TPU-Accelerator-Hardware hervorragend vergüten, wie in Abschnitt 3.2 bereits betrachtet. Sicher, die üblicherweise horrenden Preise für RZ-taugliche GPU-Accelerators stellen auch für Cloud-Provider zunächst eine Investition dar, die sich jedoch auf deren Seite sehr schnell amortisiert. (v)GPU-Manager, die NVIDIA-GPUs softwaretechnisch auf (Public-)Cloud-Hypervisoren wie GCP oder AWS partitionieren könnten, waren im betrachteten Stand für Endverbraucher nicht verfügbar.

Virtualisierer

Nehmen wir VMware als Beispiel und die in den meisten Unternehmens-RZ präsenten vSphere-Systeme: Schon interessant, dass gerade das *vGPU*-Feature, das anfänglich kaum Beachtung fand, nun nur noch in VMwares höchster Lizenz-Preisklasse zu finden ist. Ohne vSphere Enterprise Plus-Lizenz läuft im Bereich »*vGPU-Computing*« (vorab: softwarebasierte Partitionierung einer GPU) rein gar nichts. Und betrachten wir den »Aufwand«, der dahintersteckt, ist die Verortung dieses Features in der höchsten Lizenz-Preisklasse ohne jedweden technischen Hintergrund, sondern rein gewinnorientiert motiviert. Denn es handelt sich auf ESXi-Seite lediglich um die Integration eines einzigen VIBs, das zudem noch von NVIDIA geliefert wird. Dagegen ist prinzipiell nichts einzuwenden, das ist freie Marktwirtschaft. Aber dennoch wäre hier seitens VMware eine etwas niedrigere finanzielle Einstiegshürde gerade für KMU durchaus wünschenswert gewesen. Aber wozu den Profit minimieren, wenn es die GPU-Quelle #1 nicht anders vormacht.

NVIDIA

Sicher, leistungsstarke GPU-Karten waren noch nie günstig, und einige Hardcore-Gamer würden wahrscheinlich lieber eine Hypothek aufnehmen und/oder Organe spenden, als das neue Spiel XYZ nicht in Ultra-Super-HD-256K-with-unbelievable-high-FPS zu spielen. Aber erst seit ihrer neu erstrahlenden Popularität im KI/ML-Bereich schießen die Preise für »echte« GPU-Accelerators wie NVIDIAs Ampere- oder Tesla-Modelle (oder RTX-Karten für ML-Workstations) durch die Decke – fünfstellige Beträge für RZ-taugliche Karten der Oberliga sind hier keine Seltenheit, sondern das Normalprogramm, und komplette DGX-Systeme liegen wie bereits betrachtet noch deutlich darüber. Wer denkt, dass sein Unternehmen mit gemieteten, cloudgehosteten, RZ-tauglichen GPUs auf Dauer günstiger fährt – eine nette Idee, die oft nur dann aufgeht, wenn es sich nicht um 24/7-Dauerbetrieb der Systeme handelt.

4.2 Partitionierte GPUs mit NVIDIAs vGPU und MIG

Betrachten wir an dieser Stelle zunächst einige grundlegende Begrifflichkeiten rund um die möglichen Bereitstellungsarten RZ-tauglicher GPU-Hardware, da sich diese zum Teil grundlegend von »normalen« Verfahren im HEDT/Workstation-Bereich unterscheiden und wichtig für das spätere Verständnis der durchzuführenden Setup-Tasks sind.

Ein wichtiger Aspekt ist dabei die Möglichkeit, GPUs zu *partitionieren* und damit ihre immense Rechenleistung, die sonst gegebenenfalls nicht vollumfänglich genutzt wird, an multiple Client-Prozesse zu verteilen. Dazu stehen verschiedene Verfahren zur Verfügung, die wir im Folgenden betrachten.

Eine weitere Variante, die seit dem GPU-Operator 1.11 zur Verfügung steht, um Däumchen drehende GPUs in KI/ML-Clustern zu vermeiden, ist die neue Option des GPU-Sharings/Overcommitments (z. B. physikalisch 1 GPU, aber *n* Client-Prozesse dürfen die GPU beanspruchen). Dies wird konkret ab Abschnitt 10.10 betrachtet.

4.2.1 Vorbetrachtungen und Scope

Die primär für grafische Anwendungsfälle gedachten GPU-Modi *vSGA* und *vDGA* werden im Folgenden nicht angesprochen, da sie für den Scope des Buches irrelevant sind.

Für skalierbaren Datacenter-Betrieb mit Fokus auf Compute-Anwendungen (KI/ML) und multiple Tenants sind üblicherweise nur die Modi *vGPU* (vGPU = virtual GPU, »Software«-partitionierbare GPUS) und *MIG* (MIG = Multi Instance GPUs = »Hardware«-partitionierbare GPUs, die üblicherweise auch den vGPU Modus unterstützen) geeignet und von Interesse, in wenigen Ausnahmenfällen auch ein PCI-Passthrough von ganzen GPUs ohne Partitionen.

Daher liegt der Fokus aller folgenden Betrachtungen auf den GPU-Bereitstellungsmodi vGPU und MIG und den damit verbundenen Verfahren und Konzepten für einen effizienten Datacenter Betrieb.

4.2.2 Vorbetrachtungen: Partitionierte GPUs mit vGPU und MIG

Die Partitionierung einer physikalischen GPU ist besonders effizient für Workloads, die die Rechenkapazität der physikalischen GPU nicht vollständig auslasten würden. Je nach Leistungsklasse der GPU und dem konsumierenden Workload wird in der Praxis daher oftmals keine vollständige GPU benötigt. Dies gilt insbesondere dann, wenn es sich um kleinere PoC/Modelle oder Demos handelt, die auf einem Boliden wie einer A100 mit 40 oder gar 80 GB GPU-RAM betrieben werden sollen.

Ohne die Möglichkeit einer GPU-Partitionierung wird jeder ML-Workload die komplette GPU okkupieren, unabhängig davon, ob er sie tatsächlich verwendet bzw. voll auslastet … oder eben nicht. Insbesondere containerisierte Compute-Workloads im RZ/Cloud-Umfeld auf Kubernetes-/OpenShift-Clustern profitieren von der höheren Effizienz einer GPU-Partitionierung sowie der im Fall von MIG damit einhergehenden, echten Mandanten-Fähigkeit bzw. Multi-Tenancy und damit höheren Sicherheit.

NVIDIAs vGPU-(Virtual-GPU-)Software ermöglicht bereits seit langem die softwarebasierte/temporäre Partitionierung von geeigneten NVIDIA-GPUs und unterstützt auch eine breite Palette an weit verbreiteten und relativ kostengünstigen Datacenter-GPUs, wie z. B. Tesla T4-Instanzen. Für den vGPU-Modus wird auf Hypervisor-Ebene ein lizenzpflichtiger GPU-Manager benötigt.

Der *MIG-Mode (Multi-Instance GPU)*, der mit NVIDIAs Ampere-GPU-Generation und Modellen wie der A100 oder A30 erstmals zur Verfügung stand, ist zwar ebenfalls ein Betriebsmodus um GPUs zu partitionieren, aber im Gegensatz zum vGPU Mode bringt dieses Verfahren eine deutlich effizientere Multi-Tenancy, die jedoch im Vergleich zum vGPU-Mode meist mit einer höheren Idle Power einhergeht, siehe Abschnitt 5.4.4. MIG-fähige Karten können in der Regel auch eine Partitionierung mittels vGPU anbieten. Die (vorab vereinfacht und abstrakt ausgedrückt) GPU-Core-Tenancy ist dann jedoch nicht mehr aktiv. Alle Details zum MIG-Modus finden Sie ab Abschnitt 4.4.

Während sich im *vGPU-Modus* (Time-Sliced, CPU-like Scheduling) alle GPU-Partitionen im gleichen Hardware-Pool der GPU-Cores-Engine bedienen und nur den GPU-RAM partitionieren, sorgt der *MIG-Mode* für explizite Separierung der multiplen (im betrachteten Stand bis zu sieben) GPU-Instanzen. Dadurch können Workloads besser aufgeteilt und zudem die Datensicherheit erhöht werden. Sie hierzu auch: *https://www.nvidia.com/de-de/technologies/multi-instance-gpu/*

MIG-fähige Karten werden von NVIDIAs GPU-Operator ab Version 1.7 unterstützt.

4.2.3 NVIDIAs vGPU und (leider noch kein) Cloud-Einsatz

Für alle folgenden Betrachtungen ist es vorab essentiell wichtig zu verstehen, dass *NVIDIAs vGPU-Software* (der GPU-Manager) auf gängigen Public-Cloud-Plattformen wie GCP, AWS oder Azure typischerweise *nicht* zur Verfügung steht. Der GPU-Manager (und damit vGPU) steht nur für Hypervisoren zur Verfügung, wie sie in Unternehmens-RZ bzw. Private Clouds eingesetzt werden, wie z. B. vSphere, Citrix, RHEL KVM, Nutanix und andere.

Hintergrund: Der *GPU-Manager* wird bereits auf der Ebene des Hypervisors zwingend für vGPU benötigt, damit über ihn GPUs softwaretechnisch partitioniert und an die VMs durchgereicht werden können. So viel Zugriff kann oder will im betrachteten Stand keiner der Public-Cloud-Riesen dem Endkunden anbieten. In logischer Konsequenz existiert daher (bisher) auch keine GPU-Manager-Software seitens NVIDIA für AWS, GCP und Co. Somit bleiben den Unternehmen, die GPU-Workloads in der Cloud prozessieren möchten oder müssen, nur zwei Alternativen.

Entweder werden nicht-MIG-fähige Karten, wie z. B. T4, A10 oder A40 komplett (en bloc, d. h. 1:1) via Passthrough direkt an den Consumer (die KI/ML-Applikation) durchgereicht oder es kommen – wenn Tenancy gefragt ist und/oder multiple Consumer gleichzeitig auf einer großen, leistungsfähigen GPU wie der A100 sicher bedient werden sollen – MIG-fähige Karten zum Einsatz. Diese werden vom Cloud-Provider typischerweise per Passthrough durchgereicht, können aber vom Kunden bzw. dem GPU-Operator-Stack zumindest auf dem betreffenden Node in GPU-Slices partitioniert und damit von *n* Applikationen verwendet werden.

Auch wenn die Passthrough/MIG-Only-Diktatur der Cloud-Provider die Konfigurationsflexibilität deutlich einschränkt – einen kleinen Vorteil für den Endkunden bietet die vGPU-Abstinenz im Cloud-Umfeld dennoch: Die vGPU-Lizenzkosten entfallen.

4.3 vGPU – Virtual GPU

Schauen wir nun etwas tiefer in Konzepte und Arbeitsweise der vGPU-Partitionierung. Im Folgenden wird zunächst betrachtet, was genau unter vGPU zu verstehen ist, welche Produkte NVIDIA in diesem Bereich anbietet und wie vGPU konzeptionell und technisch betrachtet arbeitet.

4.3.1 Generelle vGPU-Architektur

Betrachten wir zunächst die generelle vGPU-Architektur anhand Abbildung 4.1, die eine generelle, vereinfachte Darstellung der Arbeitsweise einer vGPU und ihre Aufteilung an multiple VM-Instanzen zeigt.

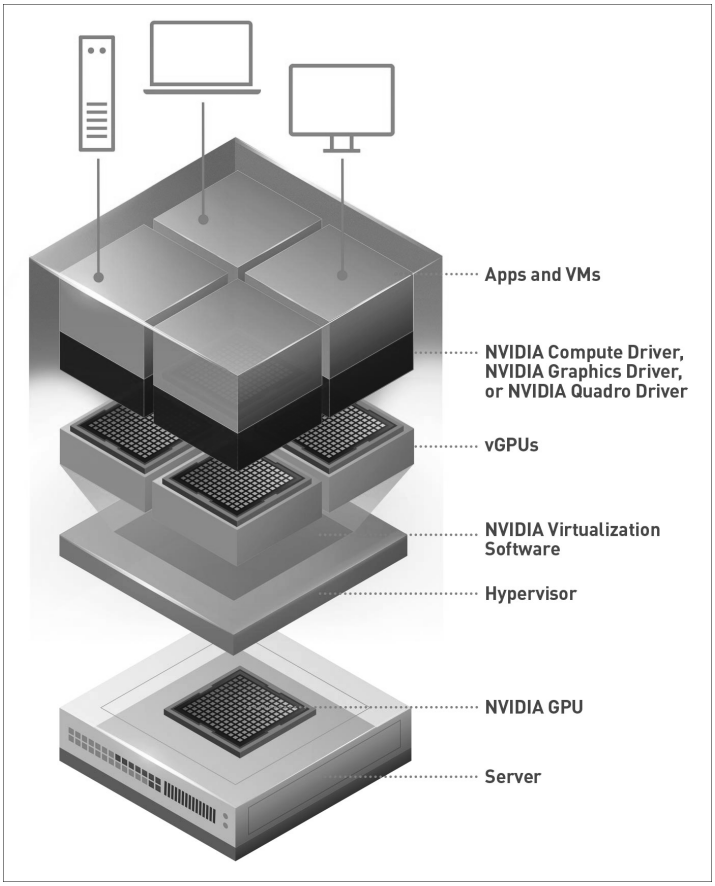


Abbildung 4.1 NVIDIAs vGPU-Stack in stark vereinfachter, konzeptioneller Form, Quelle: NVIDIA

4.3.2 Details zur Funktionsweise

Wie in Abbildung 4.1 bereits zu erkennen ist, wird die vGPU-Funktionalität als zusätzliche Software-Komponente zwingend über den jeweils verwendeten *Hypervisor* bereitgestellt. Im betrachteten Stand wurden u. a. folgende (in der Regel Typ1-)Hypervisoren/Bare-Metal-OS unterstützt (Nennung hier ohne Versionen): vSphere, OpenStack, Windows Server/Hyper-V, RHEL KVM/RHV, XenServer.

Es ist ebenfalls zu beachten, dass Ampere-Karten wie z. B. die A30 oder ihr größeres Pendant, die A100 oder die neue H100, die per *MIG* (siehe ab Abschnitt 4.4) auch hardwarebasiert (korrekter wäre »räumlich«/»EEPROM-like«) partitioniert werden können, ebenfalls rein softwarebasiert im *vGPU*-Mode betrieben werden können. Wann welcher Modus (vGPU, MIG oder MIG-backed vGPU) zum Einsatz kommt, bestimmen üblicherweise die (gegebenenfalls vorgegebene) RZ-Architektur, Security-Anforderungen und der konkrete, technische Einsatzzweck.

Bleiben wir zunächst bei der *vGPU-Partitionierung*. Jede GPU-Karte stellt eine bestimmte Anzahl an Rechenkernen und verfügbarem Grafikspeicher (RAM) zur Verfügung. Diese Compute-Ressourcen können nun an eine VM (oder eben partitioniert an mehrere) durchgeschleift werden. Während es sich bei den – von der jeweiligen VM abgerufenen – (v)GPU-Compute-Ressourcen wie üblich um eine komprimierbare Compute-Ressource handelt (vergleiche CPU/GPU-Shares), stellen die RAM-Ressourcen wie üblich ein nicht komprimierbares Hard-Limit dar. Ein einfaches Rechenbeispiel:

Reichen wir eine vGPU-fähige Karte mit 16 GB RAM an vier VMs bei identischer Partitionierung durch, so erhält jede VM exakt 4 GB GPU-RAM. Damit sind die nicht komprimierbaren bzw. nicht over-commit-fähigen Compute-Ressourcen dieser GPU-Karte erschöpft.

Achtung: vGPU und GPU-Pool-Sharing

Dabei ist jedoch zu beachten, dass die partitionierten GPU-Slices auf einer nicht-MIG-, sondern »nur« vGPU-fähigen Karte nicht wirklich voneinander isoliert sind. Alle vGPU-Partitionen z. B. einer Tesla T4 oder A10 nutzen die GPU-Ressourcen wie einen »Shared«-Pool. Wer zwingend echte *Tenancy* braucht, wird um MIG-fähige Karten nicht herumkommen.

Das vGPU-Modell bietet dabei die funktionalen Eigenschaften, die Sie in Abbildung 4.2 sehen.

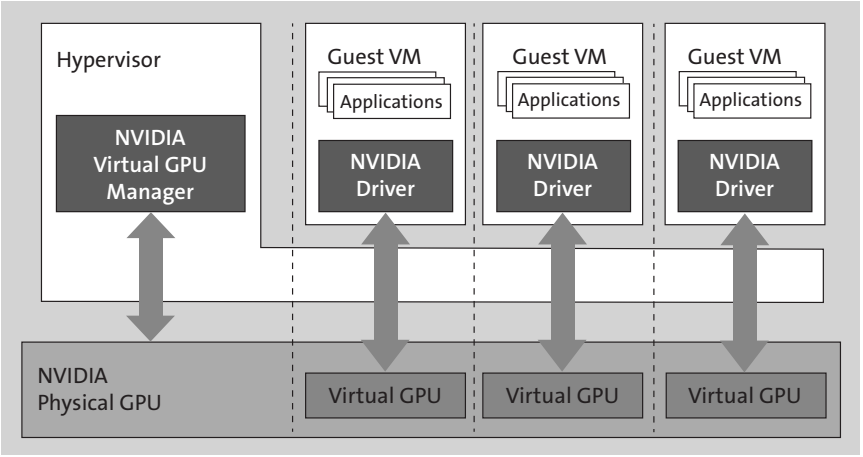


Abbildung 4.2 NVIDIA-vGPU-Systemarchitektur – High Level, Quelle: NVIDIA

Jede NVIDIA-*vGPU* kann analog zu einer herkömmlichen GPU betrachtet werden: Sie besitzt eine fixe/statische Menge an GPU-Framebuffer (RAM der GPU-Karte) und einen oder mehrere virtuelle Displayausgänge oder auch Heads.

Die virtuelle GPU (*vGPU*) erhält zum Zeitpunkt ihrer Bereitstellung exklusiv einen Teil des Framebuffers der realen GPU (oder je nach Profil gegebenenfalls auch den ganzen). Diesen verwendet die virtuelle GPU so lange, bis sie deaktiviert wird (z. B. Shutdown der VM).

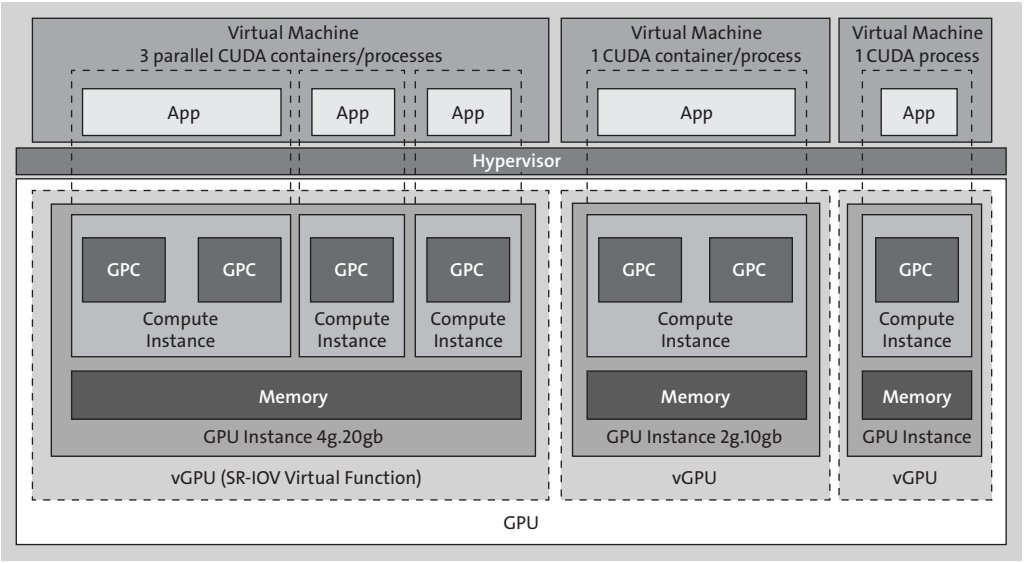


Abbildung 4.3 NVIDIA-vGPU-Systemarchitektur – Details, Quelle: NVIDIA

4.3.3 Time-Sliced vGPU

Der vGPU-Modus (ohne explizite Abschottung im GPU-Engine-Layer) ist mit jeder NVIDIA-vGPU-fähigen Karte verfügbar (siehe Abbildung 4.4).

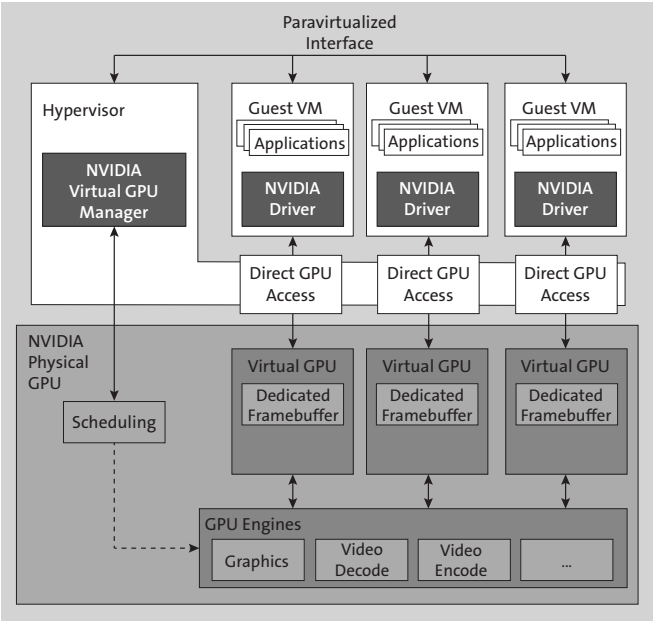


Abbildung 4.4 NVIDIA-vGPU-Systemarchitektur – Time-Sliced, Quelle: NVIDIA

vGPU-fähige Karten

Eine Auflistung vGPU-fähiger Karten von NVIDIA findet sich unter anderem in der Dokumentation NVIDIAs, wird aber auch noch in anderen Abschnitten des Buches thematisiert (u. a. Abschnitt 5.3.1): *https://docs.nvidia.com/grid/gpus-supported-by-vgpu.html*.

4.3.4 Passthrough GPU vs. vGPU im Hypervisor

Abbildung 4.5 zeigt im direkten Vergleich, wie GPUs direkt per Passthrough oder per vGPUs auf einem vSphere-System an die darüberliegenden virtuellen Clients durchgereicht werden.

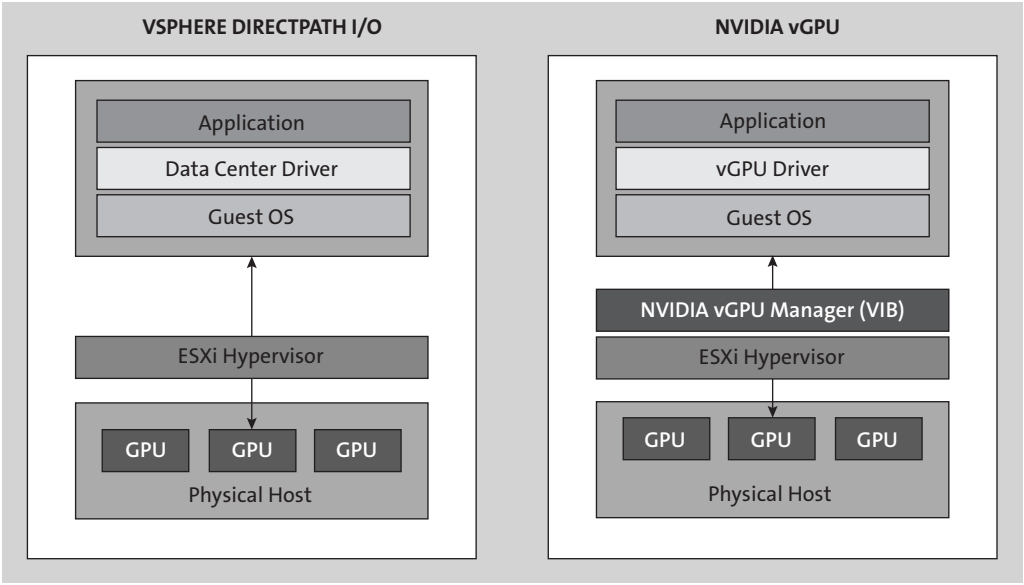


Abbildung 4.5 Passthrough GPU vs. vGPU-Workflow zwischen VM und ESXi-Hypervisor, Quelle: VMware

Im Passthrough-Mode (links) wird das PCI-Device über den Hypervisor dediziert an einen Consumer (sprich: VM) durchgereicht. Soll stattdessen der vGPU-Mode (rechts) zum Einsatz kommen, wird auf dem Hypervisor ein GPU-Manager in Form eines VIB installiert. Dieses registriert die verbauten NVIDIA-GPUs und stellt bei der Einrichtung einer VM oder eines Templates die Möglichkeit zur Verfügung, nur bestimmte Partitionen einer GPU an eine VM durchzureichen. In der VM muss bei diesem Setup-Typ auch ein passender, vGPU-fähiger Treiber installiert werden.

Diese Installation der GPU-Treiber (egal ob normaler Driver oder vGPU) erfolgt in containerisierten Umgebungen üblicherweise vollautomatisch über den GPU-Operator, der ab Abschnitt 9.4 besprochen wird.

4.3.5 vGPU-Produkte

Die Rubrik *vGPU* umfasste gemäß der Definition von NVIDIA im Stand 07/2022 die folgenden Produkte:

- *NVIDIA Virtual Applications* (*vApps*) – Dieses vGPU-Produkt ist z. B. für Unternehmen gedacht, die Citrix Virtual Apps und Desktops bereitstellen oder RDSH oder andere App-Streaming- oder sessionbasierte Lösungen einsetzen. Generell: für Anwendungen auf PC-Ebene und serverbasierte Desktops geeignet.
- *NVIDIA Virtual PC* (*vPC*) – Für Anwendungsfälle, in denen virtuelle Remote-Desktops (»virtuelle PCs«) mit PC-Windows-Anwendungen, Browsern und High-Definition-Video benötigt werden.
- *NVIDIA RTX Virtual Workstation* (*vWS*) – Für den Einsatzfall, remote grafische Systeme (z. B. CAD/CAE) mit maximaler Leistung zu betreiben.
- *NVIDIA Virtual Compute Server* (*vCS*) – Der von uns primär betrachtete Anwendungsfall: HPC-/KI-/ML-/Deep-Learning-Workloads ohne grafische Ausgabe (*Display-less*), die z. B. in VMs/Containern auf Hypervisoren (vSphere, KVM etc.) mit ECC-Unterstützung betrieben werden können.

Siehe dazu auch: *https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/solutions/resources/documents1/Virtual-GPU-Packaging-and-Licensing-Guide.pdf*.

4.3.6 vGPU-Versionen

Es ist essenziell wichtig zu verstehen, dass *NVIDIAs vGPU-Software-Version* (im betrachteten Stand 08/2022: vGPU-Version 14.2) immer der Dreh- und Angelpunkt für alle – von dem jeweiligen Release – unterstützten Features ist. Sie allein entscheidet, welche Version des *vGPU-Managers* (Hypervisor) und des *Linux-* und *Windows-Drivers* unterstützt wird und welche jeweiligen Features auf welcher GPU-Hardware und Hypervisor-Plattform supportet werden:

- *https://docs.nvidia.com/grid/index.html*
- *https://docs.nvidia.com/grid/latest/product-support-matrix/index.html*

LTS und (Short-Term) Production Branch

Wie in allen produktivtauglichen Konzepten bietet NVIDIA neben *Short-Term Releases* (unverständlicherweise von NVIDIA als *Production Branch* gekennzeichnet) auch *LTS Releases* (*Long Term Support Branch*). Leider zieht NVIDIA hier keinen »roten Faden« bezogen auf die Nummerierung (etwa durch gerade und ungerade Nummern) durch.

Einige der aktuelleren, ungeraden Release-Nummern werden für die LTS-Versionen verwendet (11, 13), einige der geraden (12, 14) für die Short-Term Branches. In älteren Versionen passt dieses Schema nicht, daher kontrollieren Sie immer den verwendeten Branch gemäß der Support-Matrix.

Ein vGPU *LTS Branch* hatte im betrachteten Stand einen Supportzeitraum von drei Jahren: So ist der EOL für den vGPU 13.x Branch, der im August 2021 gestartet wurde, im August 2024 erreicht. Die *Short-Term Branches* hingegen haben in der Regel nur einen maximalen Supportzeitraum von einem Jahr.

Hinweis: NVIDIA AI Enterprise (NVAIE-Versionen)

Bitte beachten Sie bereits an dieser Stelle, dass sich vGPU- und NVAIE-Software (Letztere ab Kapitel 6 erläutert) in Art, Aufbau und Lizenzierung (siehe ab Kapitel 7) unterscheiden. Die von NVAIE unterstützten Produkte und Features finden sich hier:

https://docs.nvidia.com/ai-enterprise/latest/release-notes/index.html.

4.3.7 vGPU-Features je nach Produkt

Tabelle 4.1 listet die vGPU-Features je nach gewähltem/lizenziertem Produkt auf. Bezogen auf den Scope des Buches ist für uns primär nur die letzte Spalte der Tabelle (vCS, virtual Compute Server) von Relevanz.

Feature	NVIDIA vApps	NVIDIA vPC	NVIDIA vWS	NVIDIA vCS
License Entitlement				
Concurrent User (CCU)	✓	✓	✓	
Per GPU				✓
Capability Entitlement				
Desktop Virtualization		✓	✓	
RDSH App Hosting	✓	✓	✓	
RDSH Desktop Hosting	✓	✓	✓	
Compute Virtualization			✓	✓
Windows Guest OS Support	✓	✓	✓	

Tabelle 4.1 vGPU-Features und -Lizenzen

Feature	NVIDIA vApps	NVIDIA vPC	NVIDIA vWS	NVIDIA vCS
Linux Guest OS Support		✓	✓	✓
Maximum Displays	1	4	4	1
Maximum Resolution	1280 × 1024	5120 × 2880 (5K)	7680 × 4320 (8K)	4.096 × 2160 (4K)
NVIDIA RTX Enterprise Software Features			✓	
OpenGL, DirectX and Vulkan	✓	✓	✓	In-situ Graphics only
CUDA and OpenCL Support			✓	✓
ECC and Page Retirement			✓	✓
Multi-vGPU			✓	✓
NVLink			✓	✓
GPU Passthrough Support	✓		✓	✓
Bare Metal Support	✓		✓	
vGPU Profile Sizes Supported				
512 MB		✓	✓	
1 GB	✓	✓	✓	
2 GB	✓	✓	✓	
3 GB	✓		✓	
4 GB	✓		✓	✓
5 GB				✓
6 GB	✓		✓	✓

Tabelle 4.1 vGPU-Features und -Lizenzen (Forts.)

Inhalt

TEIL II Implementierung von skalierbaren KI/ML-Infrastrukturen

5 Implementierung: vSphere als Hypervisor für skalierbare ML-Infrastrukturen 153

8 Kubernetes-basierte Plattformen für skalierbare, GPU-Accelerated KI/ML-Cluster 209

9 Preflights für GPU-Accelerated Container-Cluster: Operatoren 241

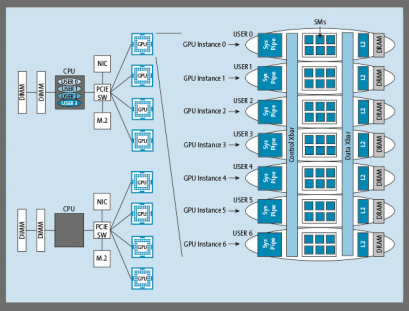
13 ML-Pipeline- und AI-End-to-End-Implementierungen mit Kubeflow/Vertex AI, Open Data Hub und NVIDIA AI Enterprise 411

Skalierbare KI/ML-Infrastrukturen

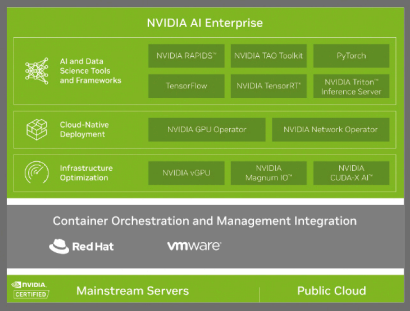
»Ready for Production«

Container-Cluster für Machine Learning und KI

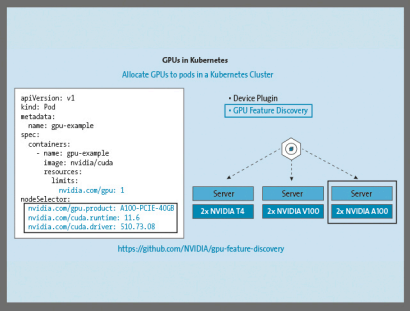
KI/ML-Modelle brauchen leistungsstarke Infrastrukturen. Oliver Liebel zeigt Ihnen, wie Sie resiliente, hochautomatisierte und autoskalierbare Systeme für Produktiv-Workloads auf Basis von NVIDIAs Datacenter-GPUs aufbauen und sie Operator-gestützt mit Kubernetes bzw. OpenShift verwalten.



Infrastrukturen planen



Komponenten verstehen



Workflows automatisieren

Best Practices für Planung und Umsetzung

Wenn große Datenmengen verarbeitet werden müssen und teure Investitionen anstehen, ist durchdachte Planung unabdingbar. Mit Hintergrundwissen und fundierten Markteinschätzungen treffen Sie richtige Entscheidungen für zukunftssichere KI/ML-Projekte und haben alle Kostenfaktoren im Blick.

Setzen Sie auf Industriestandards

In enger Zusammenarbeit mit NVIDIA wurden die vorgestellten Setups getestet und evaluiert. Sie profitieren von dieser Praxiserfahrung und erfahren, wie Sie GPUs in Kubernetes- und OpenShift-Cluster integrieren.

Zuverlässig und automatisiert skalieren

Operator-gestützte Vollautomation ist ein Muss, um KI/ML-Infrastrukturen skalierbar, resilient und kosteneffizient zu betreiben. Zahlreiche Beispiele führen Sie durch praxisnahe Konfigurationen, die für aktuelle und zukünftige Anforderungen gerüstet sind.

Getestete Setups zum Download

Linux-Enterprise-Experte **Oliver Liebel** ist Partner von Red Hat und SUSE und arbeitet eng mit NVIDIAs EGX-Team zusammen. Er ist seit 28 Jahren als Berater und Systemarchitekt tätig und hat zahlreiche GPU-beschleunigte Container-Cluster auf Kubernetes-/OpenShift-Basis für große Unternehmenskunden entworfen und umgesetzt.

Aus dem Inhalt

Rheinwerk Computing

9 783836 273930

€ 79,90 [D]
€ 82,20 [A]

ISBN 978-3-8362-7393-0
Netzwerke/Server