

---

# Skalierbare und flexible Arbeitsumgebungen für Data-Driven Sciences

Dennis Schridde<sup>1</sup>, Martin Baumann<sup>2</sup>, Vincent Heuveline<sup>3</sup>

1,2,3 Universitätsrechenzentrum, Universität Heidelberg

**Abstract.** In vielen wissenschaftlichen Fachdisziplinen stellt die IT-gestützte Analyse und Exploration erzeugter Daten einen wesentlichen Schritt zum Erkenntnisgewinn dar. Aufgrund des technischen Fortschritts, z.B. in Bezug auf Messungen wie auch auf computergestützte Simulationen, nimmt die auszuwertende Datenkapazität rasant zu. Gleichzeitig steigt die Komplexität der eingesetzten Analysemethoden. Beides zusammen führt zu einer enormen Herausforderung, die durch die IT-Systeme erfüllt werden muss.

In der Praxis werden häufig Daten auf zentralen IT-Systemen erzeugt und gespeichert, jedoch auf lokalen Workstations analysiert und visualisiert. Die Übertragung der umfangreichen Daten führt zu langen Wartezeiten. Außerdem steht lokal meist nicht die gleiche Rechenleistung wie in zentralen IT-Systemen an Rechenzentren zur Verfügung. Andererseits haben zentrale Rechensysteme oft eine hohe Einstiegshürde.

Der im Projekt bwVisu verfolgte Ansatz setzt auf Virtualisierungstechniken, die es Nutzern ermöglicht, die individuell benötigten Werkzeuge zur interaktiven Visualisierung auf einer leistungsfähigen und skalierbaren Infrastruktur zu betreiben. Nutzer können Community-spezifische Arbeitsumgebungen einsetzen, weiterentwickeln und diese wiederum mit anderen teilen. Die Bedienung der Infrastruktur erfolgt ebenfalls über eine graphische Nutzeroberfläche. Die vorgestellte Lösung ist an einen lokalen Speicherdienst für wissenschaftliche Daten angebunden, sodass Forscher in ihrer individuellen Arbeitsumgebung sehr schnell auf die Daten ihrer Projekte zugreifen können und diese sehr performant analysieren und visualisieren können.

**Schlagwörter.** Visualisierung, Remote-Visualisierung, Landesdienst

## Einführung

### Motivation und Bedarf

In vielen Fachbereichen stellt die Erhebung und Auswertung großer wissenschaftlicher Daten einen essentiellen Bestandteil im Forschungsvorgang dar. Auf Grund des technischen Fortschritts können immer schneller immer größere Datenmengen erzeugt und gespeichert werden, beispielsweise durch verbesserte Sensorik (z.B. Mikroskope oder 3D-Scanner) oder schnellere Hochleistungsrechner.

Für die Analyse solcher Daten ist häufig eine visuelle Darstellung mit interaktiver Steuerung erforderlich, die flexibel auf die jeweiligen Datensätze und Fragestellungen angepasst werden kann. Beispielsweise können komplexe dreidimensionale Strömungsvorgänge mit geeigneten Visualisierungstechniken sehr einfach erfasst und gewisse Phänomene (z.B. Wirbelstrukturen)

lokalisiert werden. Die Verarbeitung erfolgt meist in mehreren Arbeitsschritten und Phasen, in denen quantitative und qualitative Analysen durchgeführt werden (Upson et al. 1989).

Allerdings stellt die interaktive Arbeit mit solchen sehr umfangreichen Daten eine Herausforderung für die Forscher dar, da deren Übertragung auf die Workstations der Nutzer aufgrund der Größe nicht praktikabel ist: Die Länge des Kopiervorgangs steht oft in ungünstigen Verhältnis zur späteren Betrachtungsdauer oder die lokale Speicherkapazität ist nicht ausreichend. Andererseits ist die Rechenleistung von Arbeitsplatzcomputern im Vergleich zu jener von HPC-Rechenknoten häufig deutlich geringer, sodass die verarbeitbare Datenmenge stark eingeschränkt ist und die Berechnungsgeschwindigkeit für eine komfortable Nutzung nicht ausreicht (Stegmaier et al. 2003, Jomier et al. 2011, Mouton, Sons, and Grimstead 2011, Shu and Hsu 2015).

Während es zwar an einigen Universitätsrechenzentren Angebote zum interaktiven Arbeiten auf den HPC-Systemen gibt<sup>1</sup>, ist deren Nutzung häufig für die Forscher nicht zufriedenstellend, wie sich in Befragungen ergab. Zum einen hat nicht jeder Forscher Zugriff auf HPC-Ressourcen, zum anderen sind diese nicht immer zeitnah verfügbar. Gleichzeitig verfügt nicht jeder Wissenschaftler über ausreichend Erfahrung im Umgang mit HPC-Systemen und könnte diese Systeme als zu kompliziert wahrnehmen, bzw. die Anforderungen an die Arbeitsumgebung sind schwer innerhalb der Nutzungskonzepte bzw. Betriebsmodelle der IT-Systeme umsetzbar. Das typische Betriebsmodell von Hochleistungsrechnern bietet eine vordefinierte Arbeitsumgebung, bestehend aus Betriebssystem und Vorauswahl an Software und Tools, die auf Leistung optimiert wurden. Nutzer können i.d.R. zusätzliche Software für die eigene Verwendung ergänzen, manuell oder mittels eines Modul-Systems, was jedoch gewisse Linux-Kenntnisse erfordert. Die Individualisierung der Arbeitsumgebung erfolgt dabei immer nutzer- und systemspezifisch, d.h. sie muss durch jeden Nutzer auf jedem verwendeten System erneut erfolgen (Mauch et al. 2014). Außerdem sind die Nutzungszeiten bei Hochleistungsrechnern mit entsprechenden Resource-Management-Systemen auf hohe Auslastung der Systeme über größere Zeiträume optimiert, was eine spontane, interaktive Nutzung erschwert.

Wenn die Netzwerkbandbreite zwischen Datenquelle und dem lokalen Rechner des Forschers nicht genügt oder die Möglichkeiten der lokal am Arbeitsplatz vorhandenen IT-Ressourcen nicht ausreichend sind, werden die zu analysierenden Daten meist vereinfacht oder verringert (z.B. durch Reduktion der Auflösung). Dadurch kann eine Analyse zwar durchgeführt werden, allerdings kann die Datenreduktion eine große Einschränkung darstellen, da beispielsweise wichtige Merkmale nicht mehr erkennbar sein können.

Es liegen also große Datenmengen vor, die analysiert werden sollen. Gleichzeitig ist jedoch der Erkenntnisgewinn daraus aufgrund einer schwer durchführbaren Visualisierung und Analyse teilweise stark eingeschränkt. Eine Arbeitsumgebung für datenintensive Forschungsfelder sollte daher mehreren Anforderungen genügen:

1. **Interaktive 3D-Remote-Visualisierung:** Nur die Visualisierung der Daten (statt der Daten selbst) wird zum Nutzer übertragen. Eine lange Übertragungszeit der Daten kann vermieden werden. Performante Rechenressourcen (CPU und GPU) in der Nähe des Datenstandortes sind erforderlich, um z.B. aufwendige 3D-Visualisierungen zu ermöglichen.

---

1 Vgl. z.B. [https://wickie.hlr.de/platforms/index.php/Graphic\\_Environment](https://wickie.hlr.de/platforms/index.php/Graphic_Environment) oder [https://www.bwhpc-c5.de/wiki/index.php/Start\\_vnc\\_desktop](https://www.bwhpc-c5.de/wiki/index.php/Start_vnc_desktop)

2. **Nutzerfreundlichkeit:** Einfache Bedienung der Arbeitsumgebung mittels graphischer Oberfläche sollte gewährleistet sein.
3. **Nutzerspezifische Arbeitsumgebungen:** Nutzer können eigene Software in eine Arbeitsumgebung integrieren. Arbeitsumgebungen können innerhalb einer Gruppe weitergeben und gemeinsam weiterentwickelt werden.
4. **Leistung und Skalierbarkeit:** Das Potential der Hardware soll ausgeschöpft werden. Ressourcenintensive Anwendungen sollen auf mehrere Knoten verteilt werden können (horizontale Skalierbarkeit). Außerdem sollte es möglich sein, mehrere Anwendungen mit geringen Ressourcenanforderungen auf einem Knoten auszuführen (vertikale Skalierbarkeit nach unten).

### Existierende Lösungen und verwandte Arbeiten

Es gibt verschiedene Projekte, in denen mittels Virtualisierung bzw. Containerisierung flexible wissenschaftliche Arbeitsumgebungen entwickelt werden oder die eine komfortable Steuerung eines Hochleistungsrechners mittels einer Web-Oberfläche ermöglichen. Einige dieser Vorhaben werden nachfolgend vorgestellt.

In der Biologie und den Life Sciences werden teilweise bereits Container-basierte Lösungen eingesetzt, um die Verteilung von Software-Paketen und die Reproduzierbarkeit der Ergebnisse zu unterstützen (siehe z.B. „BioContainers“<sup>2</sup>). Es wurde beispielsweise bereits 2008 die iPlant Collaborative gegründet, in welcher eine Infrastruktur für die Pflanzenanalyse geschaffen wurde, um virtuelle Maschinen und später auch Container auszuführen. Diese wurde 2015 unter dem Namen „CyVerse“<sup>3</sup> einer breiteren Öffentlichkeit zugänglich gemacht. Ziel ist es, den Zugang zu leistungsfähigen IT-Ressourcen und die anschließende Analyse zu vereinfachen, wozu u.a. 2011 eine Cloud-Computing-Plattform namens „Agave“<sup>4</sup> geschaffen wurde. Das primäre Augenmerk liegt darauf, die Prozesse beginnend mit der Speicherung der Primärdaten, über deren Auswertung mittels virtualisierter Software in reproduzierbaren Umgebungen, bis hin zur Visualisierung, in einer einfach zu bedienenden Web-Oberfläche abzubilden. Während dieser Ansatz zwar die Anforderungen 2) bis 4) erfüllt, steht die interaktive 3D-Remote-Visualisierung nicht im Vordergrund. Des Weiteren ist diese Plattform stark auf die Forschungsgebiete Life Sciences und Biologie zugeschnitten. Hinzu kommt, dass viele der verwendeten Komponenten Eigenentwicklungen darstellen, für die es inzwischen etablierte Lösungen gibt.

Auch im kommerziellen HPC-Kontext gibt es Bestrebungen, die Job-Verwaltung und Ansteuerung des jeweiligen Cluster-Scheduling-Systems zu vereinfachen und mit Remote-Visualisierungs-Software zu verknüpfen. So integriert die Firma NICE mit ihrem Produkt „EnginFrame“<sup>5</sup> seit einigen Jahren die eigene Visualisierungslösung „DCV“, sowie die Lösungen anderer Hersteller (z.B. HP „RGS“, Citrix „XenDesktop“) sowie Technologien wie „VNC“ und „VirtualGL“ mit verschiedenen Job-Schedulern (Adaptive Computing „Moab“, IBM „LSF“, Univa „GridEngine“, Altair „PBS/Pro“ und „SLURM“) in einer Web-Oberfläche. Auch die Firma

---

2 <https://biocontainers.pro/>

3 <http://www.cyverse.org/>

4 <https://agaveapi.co/>

5 <http://www.nice-software.com/products/enginframe>

Adaptive Computing bietet seit ca. 2016 mit ihrem Produkt „Viewpoint“<sup>6</sup> eine Web-Oberfläche an, die die Umsetzung allgemeiner Workflows und Visualisierungsaufgaben auf einem mit dem Adaptive Computing „Moab“ Scheduling-System betriebenen traditionellen HPC-Cluster für den Nutzer vereinfachen soll. Diese beiden Ansätze bauen auf traditionellen HPC-Systemen und entsprechenden Betriebsmodellen auf. Die Flexibilität im Hinblick auf nutzerspezifische Umgebungen ist somit hierdurch festgelegt. Beispielsweise ist die Portabilität einer Arbeitsumgebung auf ein anderes System daran geknüpft, dass alle beteiligten HPC-Cluster bzgl. der zur Verfügung gestellten Software (z.B. Bibliotheken, Betriebssystem, Software-Module, etc.) homogen aufgebaut sind, sodass Nutzersoftware mit diesen HPC-Clustern kompatibel ist.

Im eng mit der wissenschaftlichen Visualisierung verknüpften Scientific-Computing-Kontext gibt es Bestrebungen, verschiedene Cloud-Technologien für wissenschaftliche Berechnungen nutzbar zu machen: So beschäftigt sich das Projekt „Singularity“ mit der Integration von Container-basierter Virtualisierung in traditionelle HPC-Umgebungen (Kurtzer, Sochat und Bauer 2017), bzw. NERSC „Shifter“ mit der Nutzung von Container-Image-basierten Distributionstechniken in HPC-Clustern (Jacobsen and Canon 2015, Bahls 2016). Hierzu muss ein gewisser Entwicklungsaufwand betrieben werden und es ist eine Abweichung von in der Industrie etablierten Techniken nötig. Auf diese Weise sollen die Vorteile der einfachen Verteilung von Software-Paketen auch hier nutzbar werden. Diese beiden Projekte konzentrieren sich auf den Aspekt der nutzerspezifischen Arbeitsumgebung (Anforderung 3).

Ebenso gibt es Bestrebungen in traditionellen HPC-Systemen verwendete Hardware, wie z.B. InfiniBand Netzwerke, in virtuellen Maschinen nutzbar zu machen (Mauch, Kunze und Hillenbrand 2013) und auch im Allgemeinen die Diskrepanz zwischen traditionellen HPC-Systemen und virtuellen Maschinen vor allem in Hinblick auf I/O-Leistung zu reduzieren. So setzte sich das „MIKELANGELO“<sup>7</sup> Projekt (Laufzeit 2015 – 2017) zum Ziel, auf diese Weise eine Nutzung von Plattformvirtualisierungstechniken mit seinen verschiedenen Vorteilen im wissenschaftlichen Kontext zu ermöglichen. Bei diesen Projekten werden sehr spezifische technische Fragestellungen adressiert, die auf die optimale Ausnutzung der Hardware-Leistungsfähigkeit im Kontext der Virtualisierung abzielen (Anforderung 4).

Gleichzeitig entwickelten sich aus Richtung der Cloud-Anbieter, angestoßen durch die Erfahrungen, die Google mit ihrem internen, „Borg“ genannten System sammelte (Verma et al. 2015), verschiedene Umgebungen zur Ausführung allgemeiner Anwendungs-Container. So existieren z.B. seit 2015 das „Datacenter Operating System“ („DCOS“) der Firma Mesosphere oder seit 2014 die „Kubernetes“ genannte Container-Orchestration-Engine der „Cloud Native Computing Foundation“ („CNCF“). Ihr Fokus ist die zeitnahe, horizontal sowie vertikal skalierbare Ausführung beliebiger Anwendungs-Container. Auch diese Projekte konzentrieren sich auf die Anforderungen 3 und 4, bieten selbst jedoch noch keine Lösungen für die Anforderungen 1 und 2.

## Unser Ansatz

Keines der zuvor genannten Projekte erfüllt alle der im Kapitel „Motivation und Bedarf“ genannten Anforderungen 1) bis 4), die aus Sicht der Autoren für eine Arbeitsumgebung für datenintensive Forschungsfelder notwendig sind. Daher besteht der Bedarf einer neuen Lösung, die in Form

6 <http://www.adaptivecomputing.com/products/hpc-products/viewpoint/>

7 <https://www.mikelangelo-project.eu/>

des Projekts bwVisu adressiert wird. Der im Rahmen dieses Projekts entwickelte Dienst setzt systematisch auf Remote-Visualisierung, wodurch das Kopieren von teilweise sehr umfangreichen wissenschaftlichen Daten vermieden werden kann. Nutzerfreundlichkeit soll durch eine sehr einfache Web-Oberfläche erreicht werden. Der Einsatz von Virtualisierungstechnologien ermöglicht nutzerspezifische Arbeitsumgebungen und Skalierbarkeit. Um die Hardware-Leistung für den Anwender nutzbar zu machen, wird in dem Projekt auf Container-basierte Operating-System-Level-Virtualisierung gesetzt.

In dieser Arbeit stellen wir den Dienst bwVisu vor, erläutern das Konzept und skizzieren die technische Umsetzung. Hierzu gehen wir insbesondere auf die weiter oben genannten Anforderungen an eine Arbeitsumgebung für datenintensive Forschungsfelder ein. Abschließend geben wir einen Ausblick auf mögliche zukünftig Erweiterungen und Verbesserungen.

## **bwVisu – ein flexibler und skalierbarer Remote-Visualisierungsdienst**

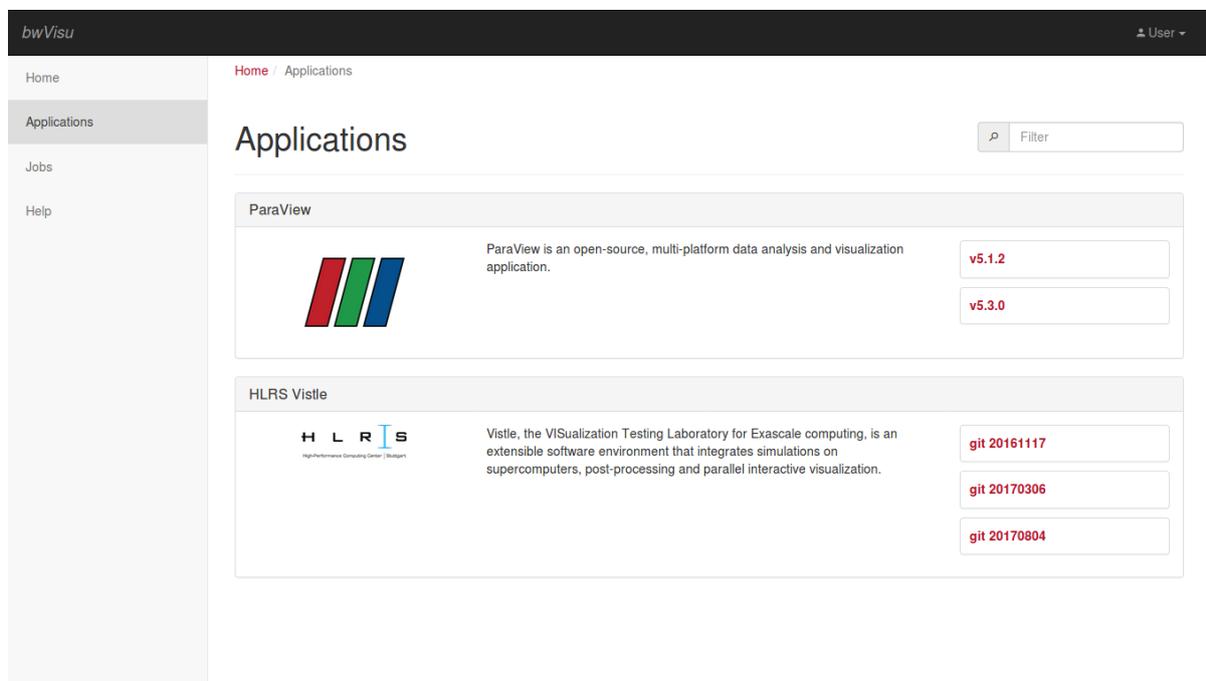
### **Übersicht**

Der Visualisierungsdienst bwVisu wird im Rahmen des gleichnamigen Projekts entwickelt, das über die Laufzeit von 2014 bis 2017 vom Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg gefördert wird. Es wurde ein Remote-Visualisierungsdienst für die Darstellung und Analyse von wissenschaftlichen Daten entwickelt, in dessen Mittelpunkt eine private Cloud-Umgebung, ausgestattet mit performanten Servern, Graphikkarten und einem Hochleistungsnetzwerk, steht. Diese wurde am Universitätsrechenzentrum Heidelberg aufgebaut, wo der Dienst auch betrieben wird. Für die Nutzer wurde ein Portal bwVisu-Web zur Ausführung individueller Arbeitsumgebungen entwickelt. Die Entwicklung und der Einsatz von Community-spezifischen Arbeitsumgebungen unterstützt Fortschritte im Bereich der „Data-Driven Sciences“ und fördert den Gedanken von „Open Data“ und „Open Source“. Die Projektpartner untersuchen gemeinsam Methoden zur verteilten wissenschaftlichen Visualisierung und entwickeln entsprechende Softwarepakete weiter, die im Rahmen von bwVisu besonders vielversprechend erscheinen. Geförderte Einrichtungen sind das Universitätsrechenzentrum Freiburg, das Universitätsrechenzentrum Heidelberg, der Lehrstuhl für Computergrafik am Karlsruher Institut für Technologie und das Höchstleistungsrechenzentrum Stuttgart. Also assoziierter Partner fungiert das Heidelberger Institut für Theoretische Studien. Das Vorhaben wird durch das Universitätsrechenzentrum Heidelberg koordiniert.

### **1. Interaktive 3D-Remote-Visualisierung**

Der Ansatz der Remote-Visualisierung sieht vor, dass die Visualisierung von Daten durch Hardware bzw. Systeme erfolgt, die einen sehr schnellen Zugriff auf die Daten ermöglichen. Die Anzeige und interaktive Steuerung der Visualisierung erfolgt jedoch von einem entfernten Rechner (also „remote“), der nur über eine vergleichsweise geringe Bandbreite mit dem Visualisierungssystem verbunden ist. Das Konzept der Remote-Visualisierung wird in vielen Werkzeugen und Umgebungen eingesetzt und ist weit verbreitet (Stegmaier et al. 2003, Jomier et al. 2011, Mouton, Sons, and Grimstead 2011, Shu and Hsu 2015). Durch diesen Ansatz können Personen auch dann

sehr umfangreiche Daten betrachten und analysieren, wenn die verfügbaren lokalen Ressourcen der Arbeitsplatzrechner oder die vorhandene Netzwerkbandbreite eher gering sind. Ein langwieriges Kopieren der zu visualisierenden Daten auf den Arbeitsplatzrechner ist nicht erforderlich. Wesentlich für diese Technologie ist, dass eine leistungsfähige Visualisierungs-Hardware vorhanden ist und diese sehr schnell auf die zu visualisierenden Daten zugreifen kann.



**Abbildung 1.** bwVisu-Web bietet dem Nutzer eine graphische Auswahl der vorhandenen Anwendungs-Software in verschiedenen Versionen mit Beschreibungstext.

Insbesondere große wissenschaftliche Daten liegen häufig bereits auf zentralen Datenspeichern vor, z.B. auf dem Speicherdienst „SDS@hd - Scientific Data Storage“<sup>8</sup> oder auf einem der Hochleistungsrechner „bwForCluster“<sup>9</sup> oder „bwUniCluster“<sup>10</sup>. Gemäß dem zuvor beschriebenen Konzept der Remote-Visualisierung werden schnelle Netzwerkverbindungen genutzt, um von den bwVisu-Systemen aus auf die Daten zuzugreifen und diese zu visualisieren. Hierbei werden möglichst sichere Zugriffswege verwendet, etwa NFSv4 mit Kerberos Authentifizierung und Verschlüsselung im Betriebsmodus „krb5p“. Durch die exzellente Anbindung an die Netzwerkinfrastruktur in Heidelberg und das Netzwerk der wissenschaftlichen Einrichtungen in Baden-Württemberg („BelWü“) sind hohe Bandbreiten zu den genannten zentralen Speicherdiensten und Speichersystemen der HPC-Cluster verfügbar. Insbesondere können die in Heidelberg lokal verfügbaren Systeme („SDS@hd“, „bwForCluster MLS&WISO“, „heiCLOUD“, etc.) mit zusätzlichen Netzwerkverbindungen mit besonders hohen Bandbreiten und niedrigen Latenzen angebunden werden.

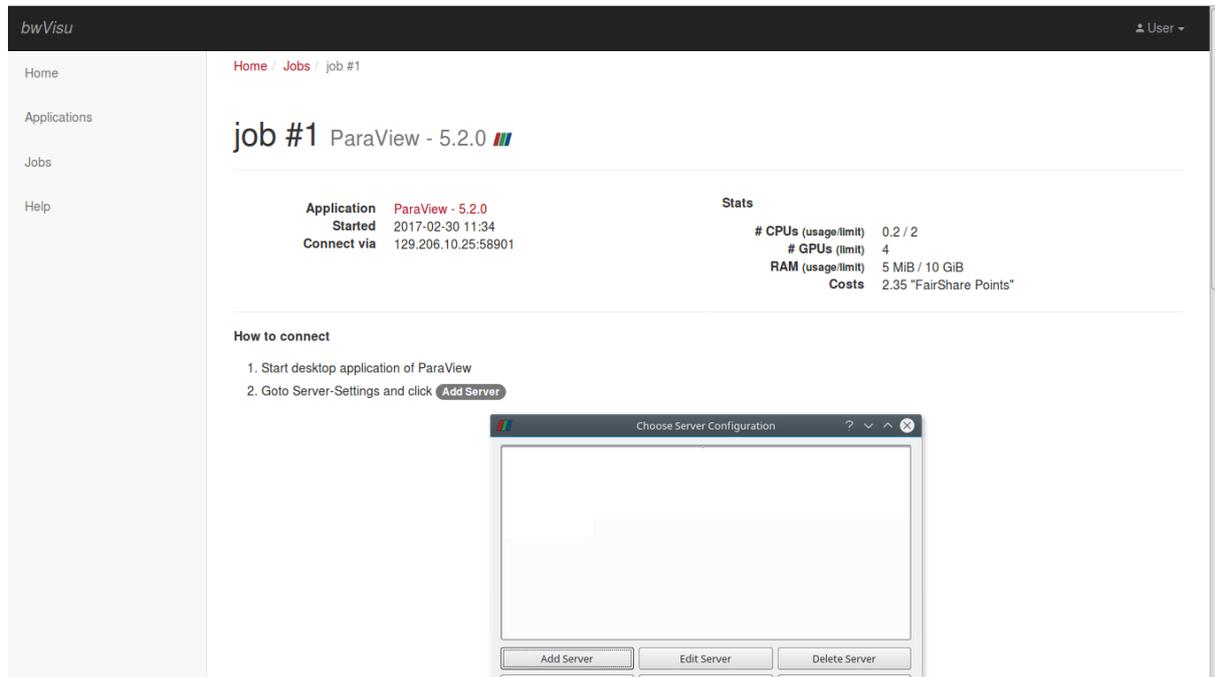
Aus dem dadurch möglichen Direktzugriff auf die Daten resultiert für Forscher die Möglichkeit noch effizienter zu arbeiten: Sie verbinden sich von ihrem Arbeitsplatz zu den Anwendungen

8 <http://sds-hd.urz.uni-heidelberg.de/>

9 [https://www.bwhpc-c5.de/wiki/index.php/BwForCluster\\_User\\_Access](https://www.bwhpc-c5.de/wiki/index.php/BwForCluster_User_Access)

10 <https://www.bwhpc-c5.de/wiki/index.php/Category:BwUniCluster>

auf der bwVisu-Hardware, von wo aus sie über die bandbreitenstarken Verbindungen des Rechenzentrums direkt auf die Daten der Speichersysteme zugreifen können.



**Abbildung 2.** Nach dem Start einer Anwendung leitet bwVisu-Web den Nutzer durch den Verbindungsvorgang und liefert nützliche Informationen.

## 2. Nutzerfreundlichkeit

Ein besonderes Augenmerk wurde auf verschiedene Aspekte der Nutzerfreundlichkeit gelegt. So sollen auch weniger technisch versierte Forscher durch eine einfache, graphische Benutzeroberfläche die Gelegenheit erhalten, das bwVisu-Remote-Visualisierungssystem zu nutzen. Insbesondere ist, im Vergleich zu vielen anderen Systemen, die Verwendung der Linux-Kommandozeile im Regelfall nicht notwendig.

Zu diesem Zweck wurde eine Web-Oberfläche (siehe 1. Abbildung) entwickelt, die es dem Nutzer ermöglicht, die gewünschte Anwendung anhand von Namen oder Beschreibungstext auszuwählen und die erforderliche Version zu starten. Nach einem weiteren Klick und Angabe weniger Informationen, wie erforderlicher Ressourcenmenge (RAM, Anzahl CPUs, Anzahl GPUs), wird die Anwendung vom bwVisu-System für den Nutzer gestartet und steht nach wenigen Sekunden zur Verfügung, siehe im Kapitel zum technischen Aufbau.

In einer Befragung ergab sich, dass Nutzer primär an einer sofortigen Nutzung eines Remote-Visualisierungsdienstes interessiert sind und eher geringen Bedarf an Reservierungen für zukünftige Nutzungen sehen. Hierauf wurde der Dienst bwVisu ausgerichtet: Nutzer verwenden freie Ressourcen und blockieren diese dadurch für andere. Die freien Ressourcen und die aktuelle Auslastung des Systems werden kontinuierlich angezeigt, sodass die Forscher ihre Ressourcen-Anforderungen ggf. hierauf anpassen können.

Sobald die Anwendung gestartet ist, erhält der Nutzer alle nötigen Verbindungsinformationen, sowie Hilfe bei der Einrichtung des evtl. nötigen Client-Programms (2. Abbildung). Für Pro-

gramme, die eine Remote-Display-Software wie „Xpra“<sup>11</sup> nutzen, kann der Zugriff zukünftig sogar ohne die gewohnte graphische Umgebung zu verlassen, direkt über die Web-Oberfläche erfolgen, da der Client direkt in die Web-Anwendung integriert sein wird. So soll erreicht werden, dass nicht nur erfahrene Nutzer, sondern sehr viele Wissenschaftler den Dienst bwVisu ohne besondere Lernphase nutzen können. Direkte Links auf die Web- und Hilfeseiten der Anwendung stehen dem Nutzer während der gesamten Nutzung zur Verfügung.

Bei dem Dienst bwVisu handelt es sich um einen Landesdienst, der u.a. von allen Forscher baden-württembergischer Hochschulen genutzt werden kann. Wie üblich bei Landesdiensten, muss der Nutzer nur über einen Account verfügen, der im Rahmen des föderierten Identitätsmanagements „bwIDM“<sup>12</sup> zur Authentifizierung und Autorisierung eingesetzt wird. Dies ermöglicht einen einfachen, einheitlichen und sicheren Zugang zu diesem Landesdienst.

### 3. Nutzerspezifische Umgebungen

Ein Hauptmerkmal von bwVisu besteht darin, dass Nutzer angepasste Arbeitsumgebungen verwenden können, die im Hinblick auf die spezifischen Anforderungen des Nutzers hin optimiert werden können. Durch die Nutzung von Container-Technik ist es für bwVisu möglich, seinen Nutzern sowohl übliche, weit verbreitete Anwendungen anzubieten, als auch äußerst fachspezifische Anforderungen zu bedienen (Higgins et al. 2015, Jacobson and Canon 2015, Bahls 2016, Priedhorsky and Randles 2016, Hale et al. 2017). Für typische Anforderungen werden durch die Projektpartner Container entwickelt und für die Nutzer des Dienstes bereit gestellt, die anschließend mit wenigen Klicks genutzt werden können. Hierzu gehören beispielsweise die weit verbreitete Open-Source-Anwendung KitWare „ParaView“<sup>13</sup> für die parallele Analyse mehrdimensionaler und zeitabhängiger Daten, sowie die am Höchstleistungsrechenzentrum Stuttgart (HLRS) entwickelte Software „VISTLE“<sup>14</sup> (VISualization Testing Laboratory for Exascale computing), aber auch die mittels Remote-Display-Software „Xpra“<sup>15</sup> zugänglichen Anwendungen UIUC „VMD“<sup>16</sup> (Moleküldynamikvisualisierung) und ISTI-CNR „Meshlab“<sup>17</sup> (3D-Dreiecksgitterbearbeitung). Diese im bwVisu-Projekt entwickelten Container-Images können durch Nutzer weiter angepasst und spezialisiert werden. Die öffentliche Weitergabe von Container-Images kann optional entfallen, z.B. um restriktiven Software-Lizenzen zu genügen.

Neben der Verwendung offizieller bwVisu-Container, besteht für Nutzer auch die Möglichkeit, eine neue Umgebung am eigenen Arbeitsplatz zu erstellen und daraufhin in bwVisu zu verwenden (3. Abbildung). Nach dem Erzeugen des Container-Images wird dieses in ein Container-Repository (Teil einer sogenannten Container-Registry) geladen. Schließlich wird die Anwendung unter Angabe einiger Metadaten zum Anwendungstyp, Autor, usw. in die bwVisu-Anwendungsdatenbank eingepflegt und steht dann zur Verwendung bereit. Nachdem dieser Vorgang einmal durchgeführt wurde, ist das Starten der Anwendung für Nutzer ohne Entwicklungserfahrung mittels der graphischen Web-Oberfläche einfach möglich.

---

11 <https://xpra.org/>

12 <https://www.bwidm.de/>

13 <http://www.paraview.org/>

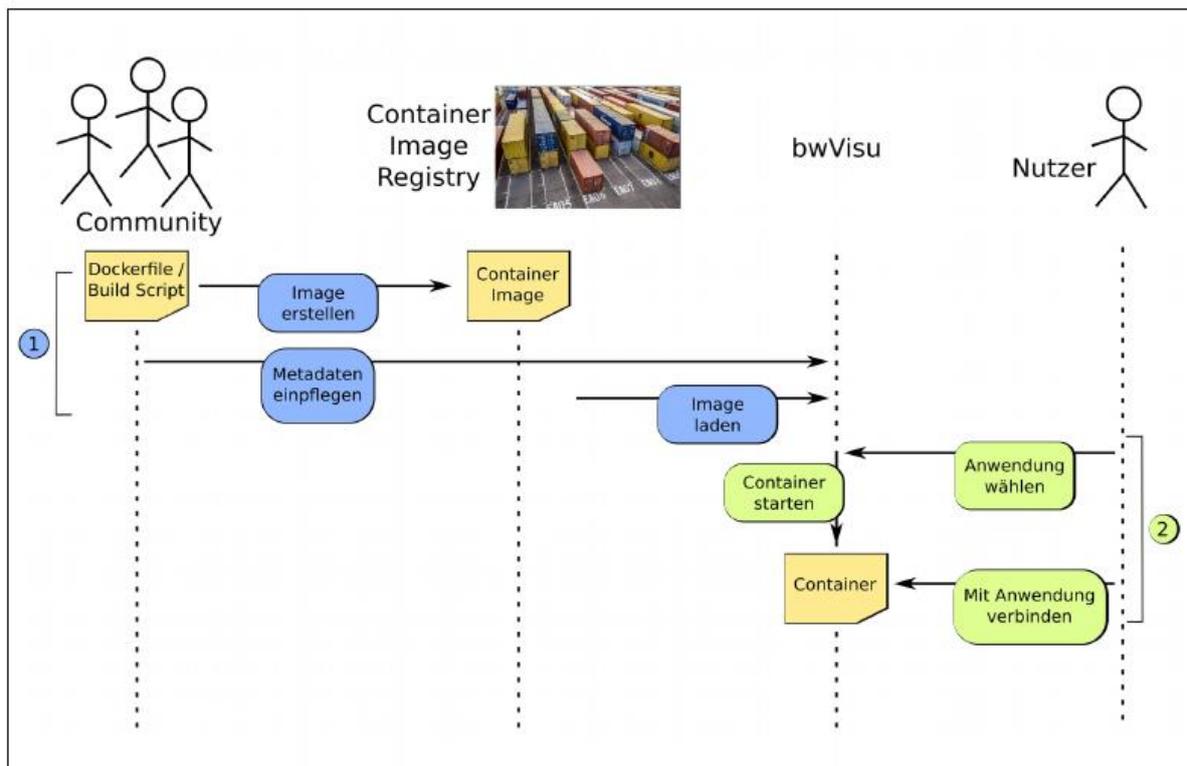
14 <http://www.hlrs.de/vistle>

15 <https://xpra.org/>

16 <http://www.ks.uiuc.edu/Research/vmd/>

17 <http://www.meshlab.net/>

Der genannte Vorgang des Erstellens eines Container-Images ist vergleichbar mit der Installation oder Kompilierung von Anwendungen auf einer Linux-Kommandozeile und erfordert gewisse Kenntnisse im Bereich Linux. Um die Einstiegshürden zu senken, haben die bwVisu-Projektpartner bereits verwendbare Grundbausteine erstellt, etwa zur Anbindung an Nutzerverzeichnisse oder Nutzung von Remote-Display-Software. Darüber hinaus soll eine Dokumentation mit Empfehlungen und Richtlinien erstellt werden, aus der hervorgeht, wie eine Anwendung mit der bwVisu-Steuerung kommunizieren kann oder wie Graphikbeschleunigung nutzbar wird. So können sich Nutzer voll auf die Installation und Konfiguration ihrer Arbeitsumgebungen und Anwendungen konzentrieren.

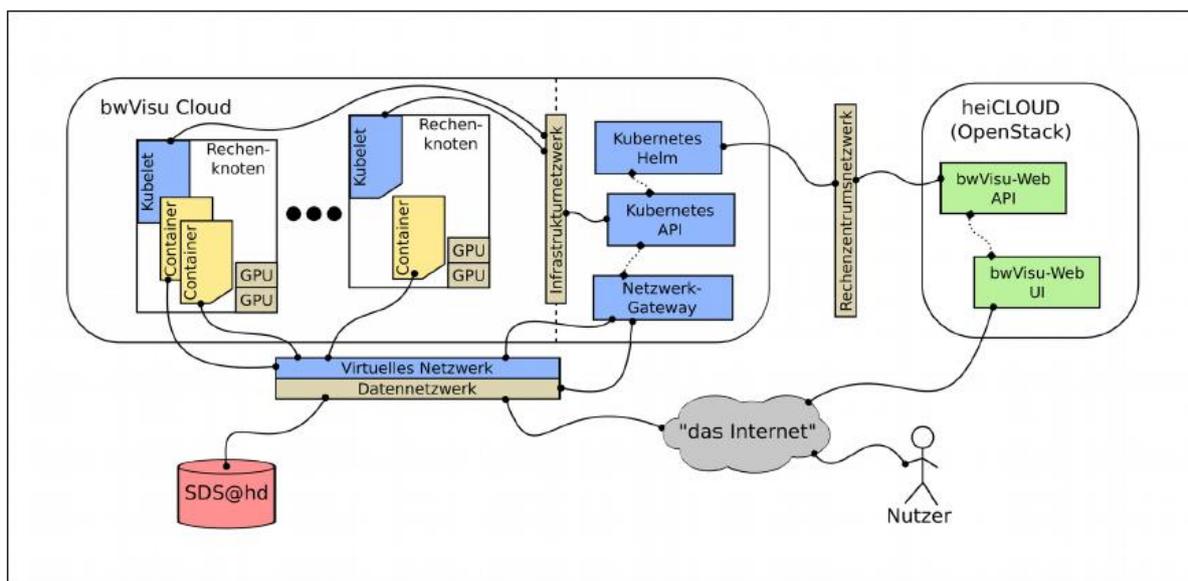


**Abbildung 3.** Nachdem die Community ein Container Image erstellt hat (1), kann sich der Nutzer nach wenigen Schritten mit seiner Anwendungsinstanz verbinden (2). (Graphik unter Verwendung von Bildmaterial unter CC BY-ND 2.0 Lizenz © 2014 Tristan Taussac: <https://www.flickr.com/photos/triantaussac/15145365916>)

Um redundante Arbeit zu vermeiden, ist eine gemeinsame Weiterentwicklung und Pflege dieser Arbeitsumgebungen im Sinne des Free-and-Open-Source-Software-Gedankens in der jeweiligen Arbeitsgruppe oder der weltweiten Forschungsgemeinschaft empfehlenswert: Da ein Container-Image nach Erstellung ohnehin auf einer Container-Registry verfügbar ist, kann es dort auf Wunsch auch mit anderen Forschern geteilt oder auf anderen Rechen-Clustern oder Visualisierungssystemen verwendet werden. Nutzer können sozusagen ihre eigene Software zu bwVisu „mitbringen“ und diese auch von dort auf andere Systeme „mitnehmen“.

#### 4. Skalierbarkeit und Leistung

Für den Dienst bwVisu wurden am Standort Heidelberg leistungsstarke Rechenknoten mit großem Arbeitsspeicher und performanten Graphikkarten beschafft. Eine gute Auslastung der vorhandenen Hardware wird durch den Einsatz von Virtualisierungstechnik und insbesondere durch vertikale Skalierbarkeit nach unten ermöglicht: Sofern Ressourcenanforderungen der Nutzer und Auslastung der Maschinen es zulassen, können mehrere Anwendungen auf einem Rechenknoten ausgeführt werden. Dies wird durch das bwVisu Ressource-Management organisiert, ohne dass der Nutzer selbst Maßnahmen treffen muss. Da Container-Technik nahezu keinen Overhead gegenüber der nicht-virtualisierten Anwendung verursacht, besteht – abgesehen von der Beeinflussung der Anwendungen untereinander – kein Nachteil gegenüber einer Exklusivnutzung von Rechenknoten (Xavier et al. 2013, Felter et al. 2015). Für die Forscher besteht weiterhin die Möglichkeit, ihre Arbeitsumgebung horizontal zu skalieren, also auf mehrere Rechenknoten verteilt auszuführen, sodass eine besonders große Rechenleistung von einer einzelnen Anwendung genutzt werden kann und besonders große Datenmengen verarbeitet werden können, sofern die Anwendung dies unterstützt. Somit kann eine hohe Skalierbarkeit und Performance bei gleichzeitiger optimaler Ausnutzung der Hardware erreicht werden.



**Abbildung 4.** Aufbau der bwVisu-Cloud. Die Steuerung der virtuellen Funktionen und Netzwerke (blau) wird durch „CNCF Kubernetes“ übernommen, welches die Container mit den Arbeitsumgebungen der Forscher (gelb) verwaltet. Der Zugriff auf die in SDS@hd gespeicherten Forschungsdaten erfolgt über ein Datennetzwerk, welches den Forschern gleichzeitig den Remote-Zugang zu den in Containern laufenden Anwendungen ermöglicht. Start und Stop derselben steuert der Nutzer über die bwVisu Web-Oberfläche (grün). Physikalisch existierende Einheiten sind braun dargestellt.

#### Technischer Aufbau der bwVisu-Cloud

Wie zuvor bereits erwähnt, kommt in bwVisu Container-Technologie zum Einsatz. Dadurch werden kurze Startzeiten der Anwendungen (s. Kapitel „Nutzerfreundlichkeit“) ermöglicht, denn der für Anwendungs-Container-Images erforderliche Speicherplatz ist im Vergleich zu Images virtueller Maschinen, die ein vollständiges Betriebssystem enthalten, sehr gering (Xavier et al. 2013,

Felter et al. 2015, Higgins et al. 2015, Jacobson and Canon 2015, Hale et al. 2017), was die Download-Zeit im Hintergrund stark verringert. Gleichzeitig muss hierdurch nur die Anwendung selbst, nicht jedoch ein vollständiges Betriebssystem gestartet werden, was die Wartezeit bis zur Nutzung einer Anwendung weiter reduziert. Für häufig genutzte Software kann die zum Start benötigte Zeit auf oft weniger als eine Sekunde gesenkt werden.

Der schematische Aufbau der bwVisu-Cloud wird in der 4. Abbildung dargestellt. Im geplanten Ausbau besteht sie aus 10 Rechenknoten mit je 128 GiB RAM und 2 CPUs à 14 Kernen. Die Hälfte dieser Knoten verfügt über je 2 Graphikkarten vom Typ AMD „FirePro S7150 X2“, wogegen die andere Hälfte mit je 2 Graphikkarten vom Typ Nvidia „Tesla M10“ ausgestattet ist. Diese Maschinen sind jeweils an einen 10 GbE Switch angebunden, der wiederum an BelWü und das Datennetz des des Speicherdienstes „SDS@hd“-Dienst angebunden ist. Entsprechend sind hohe Zugriffsgeschwindigkeiten der Anwendungen auf die Daten gewährleistet. Zusätzlich verfügt jeder Visualisierung-Server über eine Anbindung an ein bwVisu-internes InfiniBand-FDR-Netzwerk, über das zukünftig anwendungsinterne Kommunikation latenzarm mit hoher Bandbreite (56 Gbps) transportiert werden soll. Alle beteiligten Maschinen werden einheitlich mit CoreOS „Container Linux“ betrieben, einem minimalen, auf die Ausführung von Anwendungs-Containern ausgelegtem Betriebssystem.

Gesteuert werden diese Rechenknoten von mehreren redundant ausgelegten Infrastrukturknoten, auf denen u.a. die Container-Verwaltung CNCF „Kubernetes“<sup>18</sup>, die Netzwerksteuerung Tigera „Calico“<sup>19</sup>, das Monitoring-System CNCF „Prometheus“<sup>20</sup> ausgeführt werden. Diese Steuerkomponenten können zur Erhöhung der Verfügbarkeit auf der Heidelberger „OpenStack“-basierten Private-Cloud-Lösung „heiCLOUD“<sup>21</sup> betrieben werden, da die Kommunikation der Steuerungs- mit den Rechenknoten über HTTPS verschlüsselt und authentifiziert wird. Die genannten Infrastrukturdienste werden genauso wie die Remote-Visualisierungsanwendungen in Containern betrieben.

Die Steuerung des „Kubernetes“-Systems wird wiederum durch den bwVisu-API-Dienst vorgenommen, auf den die Weboberfläche bwVisu-Web zugreift. Durch diese zweistufige Architektur (Trennung von Frontend und Backend) können leicht Vereinfachungen, erweiterte Kontrollmechanismen oder zusätzliche bwVisu-spezifische Funktionen implementiert werden, da somit eine zentrale Stelle existiert, die von sämtlichen Nutzeranfragen durchlaufen wird. Dieser API-Dienst ist es auch, welcher in Verbindung mit Microsoft „Helm“<sup>22</sup> die einfache Anforderung des Nutzers nach einer bestimmten Anwendungsumgebung in Kubernetes-Begriffe wie virtuelle Netzwerke, Datenanbindungen an „SDS@hd“ und Container übersetzt und gleichzeitig die Authentifizierung des Datenzugriffs durch Bereitstellung von Kerberos-Tickets ermöglicht.

Der Remote-Zugriff des Nutzers auf seine laufenden, virtuellen Arbeitsumgebungen erfolgt über Netzwerk-Gateways, die die internen, virtuellen Netzwerke mit den externen (z.B. „das Internet“) verbinden, was z.B. durch für SDNs (Software Defined Network) ausgelegte Hardware übernommen werden kann.

---

18 <https://kubernetes.io/>

19 <https://www.projectcalico.org/>

20 <https://prometheus.io/>

21 <http://heicloud.uni-heidelberg.de/>

22 <https://helm.sh/>

## Zusammenfassung und Ausblick

Da in den unterschiedlichsten wissenschaftlichen Fachdisziplinen sehr umfangreiche Daten anfallen, die visualisiert und verarbeitet werden müssen, sind bestimmte Anforderungen an die entsprechenden Werkzeuge und Arbeitsumgebungen zu stellen: Neben rein technischen Anforderungen (Leistung und Skalierbarkeit) sind bedarfsgerecht auch die Nutzerfreundlichkeit und der Wunsch nach Community-spezifischen Arbeitsumgebungen zu betrachten. In dieser Arbeit werden diese Anforderungen vorgestellt und Projekte verschiedener Gruppen genannt, die manche dieser Anforderungen adressieren und Konzepte und technische Lösungen für Teilaspekte bieten.

Im Hauptteil der Arbeit wird der dem Projekt bwVisu zugrunde liegende Ansatz für einen Remote-Visualisierungsdienst vorgestellt. Durch den Einsatz von Containern und einem gut abgestimmten Betriebs- und Nutzungsmodell, kann die vorhandenen Rechenressourcen über eine Web-Oberfläche für die Datenanalyse und -visualisierung angeboten werden. Der Zugriff auf lokale Datenspeicher, sowie die Anbindung von entfernten Speichersystemen über existierende breitbandige Netzwerke wird im Projekt ebenfalls untersucht. Der gleichnamige Dienst bwVisu ist noch in der Entwicklung und verspricht eine bedarfsorientierte Lösung zu werden.

Im weiteren Verlauf des Projekts sollen weitere Nutzungsszenarien und Anwendungen in bwVisu integriert und die Nutzerdokumentation erweitert werden. Diese Maßnahmen sollen dazu dienen, dass neue Nutzer den Dienst besonders einfach für ihre jeweiligen Forschungsvorhaben einsetzen können. Sofern der Bedarf dies erfordert, können Warteschlangen und Reservierungen für zukünftige Ressourcen-Nutzung eingeführt werden.

Die Erstellung neuer Container-Images, sowie deren Einbindung in bwVisu, soll durch verschiedene Maßnahmen in der Zukunft noch weiter vereinfacht werden. Derzeit erfordert das lokale Testen eines Container-Images auf dem Arbeitsplatzrechner viel Vorwissen. Eine stärkere Unterstützung der Container-Erstellung durch Automatismen würde den Entwicklungsprozess vereinfachen und beschleunigen. Hierzu wären z.B. Scripte nötig, die automatisiert auf dem lokalen Arbeitsplatzcomputer eine Umgebung ähnlich der bwVisu-Cloud schaffen können, sodass der Anwender sein Image mit wenigen Befehlen lokal testen kann.

Zur Zeit werden neue Container-Images noch manuell durch Administratoren in die Anwendungsdatenbank von bwVisu eingepflegt. Dieser Vorgang könnte zukünftig graphisch unterstützt werden, sodass Nutzer diese Eintragungen selbstständig vornehmen können, und auch die Nutzungsrechte in der Arbeitsgruppe und der Forschungsgemeinde selbst verwalten können. Zusätzlich wären Automatismen hilfreich, welche neu in die Container-Repositories geladene Images automatisiert zur Verfügung stellen, oder bei der Erkennung der Anwendungsart (z.B. "MPI" oder "Xpra") unterstützen.

Zusätzlich dürfte es für Nutzer hilfreich sein, direkt und live Zugriff auf die Monitoring-Daten des bwVisu-Systems zu erhalten. Dies kann z.B. helfen, die Ressourcen-Anforderungen an die tatsächliche Nutzung anzupassen oder das Verhalten ihrer Anwendung unter Gesichtspunkten der Leistung (z.B. CPU- oder I/O-Auslastung auf Prozessebene) besser zu verstehen. Langfristig wäre auch die Möglichkeit zusätzlicher Dienstleistungen, wie integrierte Debugger oder Performance-Monitoring-Werkzeuge auf noch feingranularerer Ebene, interessant.

Datenzugriffe sind stets limitiert durch die Netzwerkverbindung zwischen der bwVisu-Cloud und dem System, auf dem sich die zu analysierenden Daten befinden. Bisher wurde der lokale Speicherdienst „SDS@hd“ an bwVisu angebunden. Da in Baden-Württemberg durch „BelWü“ eine Netzwerkverbindung von hoher Bandbreite zwischen den wissenschaftlichen Einrichtungen

bereitgestellt wird, könnten standortübergreifende Datenzugriffe im Rahmen der Analyse in Betracht gezogen werden. Beispielsweise ist im Rahmen des Projekts geplant die bwVisu-Cloud mit dem Hochleistungsrechner „bwForCluster NEMO“ in Freiburg zu verbinden, um auf die dort befindlichen Daten zugreifen zu können. Durch diese Schritte könnte es bwVisu mehr Nutzern erlauben, ortsunabhängig, insbesondere unabhängig vom Ort der Datenspeicherung, zu arbeiten.

Die Entwicklung und Erprobung des Dienstes bwVisu geschieht direkt auf dedizierter Hardware, ohne zusätzliche Virtualisierungsschichten („bare-metal“), da dies die beste Leistung bietet (Xavier et al. 2013, Felter et al. 2015, Higgins et al. 2015). Eine Integration in traditionelle HPC-Systeme wäre aus Leistungssicht ebenfalls gut motiviert, jedoch müsste aus Sicht der Infrastruktur eine Cloud-artige Flexibilität vorhanden sein. Entsprechende Möglichkeiten sollten zukünftig untersucht werden. Weiterhin könnte auch ein Betriebsmodell auf einer oder mehreren Cloud-Plattformen untersucht werden, welches voraussichtlich Automatisierungs- und Administrationsvorteile mit sich bringen könnte. Es sollte zukünftig geprüft werden, welche Nachteile bzgl. der Ausnutzung der Leistungsfähigkeit der Hardware sich daraus ergeben würden, und wie sich diese z.B. mit den Erkenntnissen aus o.g. „MIKELANGELO“ Projekt für bestimmte Nutzungsszenarien verringern lassen. Die Integration von bwVisu in vorhandene HPC- oder Cloud-Systeme birgt potentielle Synergie-Effekte bzgl. der Hardware-Auslastung. Die erforderlichen Anpassungen und Weiterentwicklungen könnten in einem Folgeprojekt adressiert werden.

## Danksagung

Das Projektvorhaben bwVisu wurde durch das Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg (MWK) gefördert.

## Literaturangaben

Bahls, Donald. "Evaluating Shifter for HPC Applications" In *Cray User Group Conference Proceedings*. 2016.

Felter, Wes, Alexandre Ferreira, Ram Rajamony, and Juan Rubio. "An updated performance comparison of virtual machines and linux containers" In *Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium On*, pp. 171-172. IEEE, 2015.

Hale, Jack, Lizao Li, Chris Richardson, and Garth Wells. "Containers for portable, productive and performant scientific computing" *Computing in Science & Engineering* (2017).

Higgins, Joshua, Violeta Holmes, and Colin Venters. "Orchestrating docker containers in the HPC environment" In *International Conference on High Performance Computing*, pp. 506-513. Springer, Cham, 2015.

Jacobsen, Douglas M., and Richard Shane Canon. "Contain this, unleashing docker for HPC" *Proceedings of the Cray User Group* (2015).

- Jomier, Julien, Sebastien Jourdain, Utkarsh Ayachit, and Charles Marion. "Remote visualization of large datasets with MIDAS and ParaViewWeb." In *Proceedings of the 16th International Conference on 3D Web Technology*, pp. 147-150. ACM, 2011.
- Kurtzer, Gregory M., Vanessa Sochat, and Michael W. Bauer. "Singularity: Scientific containers for mobility of compute." *PloS one* 12, no. 5 (2017): e0177459.
- Mauch, Viktor, M. Bonn, S. Chilingaryan, A. Kopmann, W. Mexner, and D. Ressmann. "OpenGL-based data analysis in virtualized self-service environments." *Proc. PCaPAC2014*, <http://jacow.org> (2014).
- Mauch, Viktor, Marcel Kunze, and Marius Hillenbrand. "High performance cloud computing." *Future Generation Computer Systems* 29, no. 6 (2013): 1408-1416.
- Mouton, Christophe, Kristian Sons, and Ian Grimstead. "Collaborative visualization: current systems and future trends." In *Proceedings of the 16th International Conference on 3D Web Technology*, pp. 101-110. ACM, 2011.
- Priedhorsky, Reid, and Timothy C. Randles. "Charliecloud: Unprivileged containers for user-defined software stacks." (2016).
- Shi, Shu, and Cheng-Hsin Hsu. "A survey of interactive remote rendering systems." *ACM Computing Surveys (CSUR)* 47, no. 4 (2015): 57.
- Stegmaier, Simon, Joachim Diepstraten, Manfred Weiler, and Thomas Ertl. "Widening the remote visualization bottleneck." In *Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the 3rd International Symposium on*, vol. 1, pp. 174-179. IEEE, 2003.
- Upson, Craig, T. A. Faulhaber, David Kamins, David Laidlaw, David Schlegel, Jeffrey Vroom, Robert Gurwitz, and Andries Van Dam. "The application visualization system: A computational environment for scientific visualization." *IEEE Computer Graphics and Applications* 9, no. 4 (1989): 30-42.
- Verma, Abhishek, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. "Large-scale cluster management at Google with Borg." In *Proceedings of the Tenth European Conference on Computer Systems*, p. 18. ACM, 2015.
- Xavier, Miguel G., Marcelo V. Neves, Fabio D. Rossi, Tiago C. Ferreto, Timoteo Lange, and Cesar AF De Rose. "Performance evaluation of container-based virtualization for high performance computing environments" In *Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on*, pp. 233-240. IEEE, 2013.