

# Skalierbare Analysewerkzeuge im Zeitalter der Data-Driven Sciences

Dennis Schridde

mit Martin Baumann und Vincent Heuveline

Universität Heidelberg  
Universitätsrechenzentrum Heidelberg  
Im Neuenheimer Feld 293  
69120 Heidelberg





- Universitätsrechenzentrum Heidelberg (Baden-Württemberg)
- Service-Bereich „Future IT - Research & Education“ (Forschungsprojekte)
- Arbeitsgebiet: Virtualisierung und Visualisierung

# Motivation



# Data-Driven Sciences

- *Forschung unter Verwendung großer Datenmengen*
- z.B. Hochdurchsatzmikroskopie
  - hohe Auflösung, Volumendaten
  - Durchsatz: bis zu 1 Gbps
  - typische Projektgröße: 10 TiB
- z.B. Next Generation Sequencing
  - Durchsatz: ca. 2 TiB / Woche pro Arbeitsgruppe
- z.B. numerische Simulationen
  - Hochleistungsrechnern ermöglichen hohe räumliche und zeitliche Auflösung
  - typische Projektgröße: zw. 0.1 TiB und 10 TiB (Strömungssimulation)



# Herausforderungen

Große Datenmengen stellen das Land vor neue Herausforderungen:

- Speicherung
  - Lösung: Zentrale Speicherdienste,  
z.B. SDS@hd (bald als Landesdienst verfügbar)
- Übertragung
  - Lösung: BelWü Netze (100 Gbps zw. Uni-RZ in Baden-Württemberg)
- Teilen / gemeinsame Nutzung
  - Lösung: bwIDM, Repositorien, zentrale Speicherdienste (s.o.)
- Analyse



# Herausforderungen

Große Datenmengen stellen das Land vor neue Herausforderungen:

- Speicherung
  - Lösung: Zentrale Speicherdienste,  
z.B. SDS@hd (bald als Landesdienst verfügbar)
- Übertragung
  - Lösung: BelWü Netze (100 Gbps zw. Uni-RZ in Baden-Württemberg)
  - Von RZ ins Büro i.d.R. weiterhin nur 1 Gbps
- Teilen / gemeinsame Nutzung
  - Lösung: bwIDM, Repositorien, zentrale Speicherdienste (s.o.)
- Analyse
  - Lösungsansatz: Vermeidung von Übertragung durch interaktive Remote-Visualisierung



# Interaktive Remote-Visualisierung

- *Fernsteuerung einer interaktiven Anwendung auf entferntem Rechner vom Arbeitsplatz aus*
  - ① Daten verbleiben auf zentralem Speichersystem
  - ② werden auf nahem Visualisierungssystem verarbeitet
  - ③ nur Bild wird zu entferntem Anzeigegerät übertragen
- Vorteile:
  - Geteilte Anschaffungs- und Betriebskosten
  - Größere Rechenleistung / Arbeitsspeicher
  - Bessere Netzwerkanbindung / Datenzugriff
- Neue Herausforderung:
  - Nutzer benötigen verschiedene / eigene Software

# Landesforschungsprojekt bwVisu





- Laufzeit: 2014 – 2017
- Koordination: Prof. Dr. Vincent Heuveline, Universität Heidelberg
- Projektpartner: HLRS, KIT, UFR, UHD



- Assoziierter Partner: HITS

 Heidelberg Institute for  
Theoretical Studies


- Gefördert durch: MWK, Baden-Württemberg



Baden-Württemberg

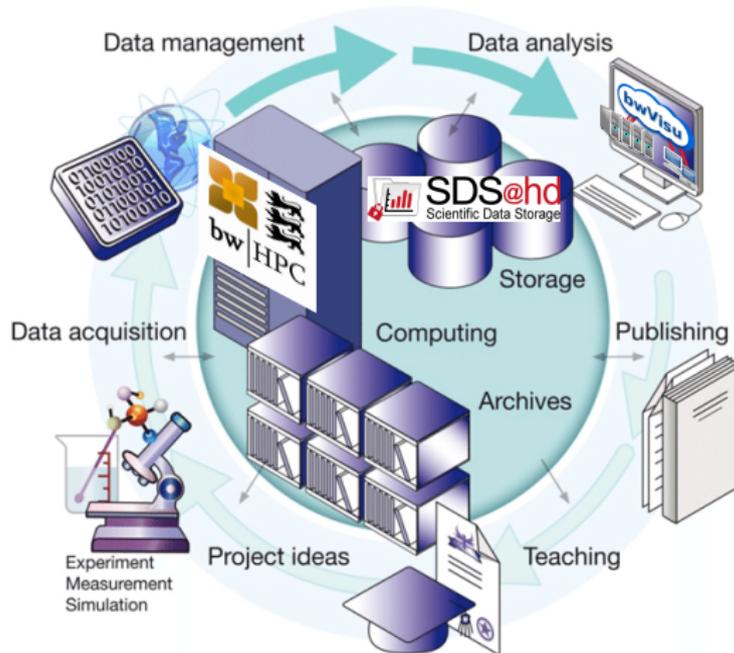
MINISTERIUM FÜR WISSENSCHAFT, FORSCHUNG UND KUNST



# Projektauftrag

Entwicklung eines Dienstes zur interaktiven Remote-Visualisierung großer, wissenschaftlicher 3D-Daten für die Forscher Baden-Württembergs basierend auf Cloud-Technologien.

# Einordnung in Datenlebenszyklus



© Jung et al., LSDMA project

# Gewünschte Eigenschaften



# Usability

Aktuell (Betriebsmodell Hochleistungsrechner):

- Remote-Visualisierung oft verfügbar, aber ...
- Technisch nicht versierte Nutzer häufig durch Kommandozeile abgeschreckt
- Können Komplexität nicht selbst reduzieren
- Wollen sich nicht mit Technikalitäten rumschlagen



# Usability

bwVisu:

- Graphischer Zugang („bwVisu Web“)
- wenige Medienbrüche (nur Web-Oberfläche und Nutzer-Software)
- Bekannte Konzepte („App Store“)
- Hilfestellung („Was muss ich als nächstes tun?“)



# Software-Distribution

Aktuell (Betriebsmodell Hochleistungsrechner):

- `./configure && make` in `$HOME`
  - Erfordert technische Fachkenntnisse
- `module load`
  - Nur von Administratoren durchführbar



# Software-Distribution

bwVisu:

- Community kann Anwendungen selbst pflegen
- Nutzung mit geringen technischen Vorkenntnissen möglich
- Einfache Nutzung verschiedener Versionen



# Skalierbarkeit

- *Leistungssteigerung durch Hinzufügen von Ressourcen*
- „Vertikal“: Arbeitskraft pro Rechenknoten
  - Limitierung durch Hardware
- „Horizontal“: Rechenknoten pro Arbeitseinheit



# Skalierbarkeit (Horizontal)

Aktuell (Betriebsmodell Hochleistungsrechner):

- Meist 1 Nutzer / Anwendung pro Knoten
  - Gut nach oben
  - Selten nach unten



# Skalierbarkeit (Horizontal)

bwVisu:

- Bessere Auslastung bei kleinen Job-Größen (wenige GiB)
  - Kürzere Wartezeiten
  - Mehr Nutzer gleichzeitig
  - Höhere Energie- / Kosteneffizienz
- Große Jobs (einige TiB) uneingeschränkt möglich

# Virtualisierung

Kann unsere Anforderungen an Software-Distribution, Skalierbarkeit und teilw. Usability lösen:

- Software in sofort ausführbaren Images
- Physikalische Maschinen können geteilt werden
- Lösungen mit HTTP API verfügbar (ideal für bwVisu Web)

# Virtualisierung

Wir nutzen Linux Container (statt virtueller Maschinen), u.A. weil:

- native I/O-Performance
- „Container Image“ Konzept ausgereift:
  - self-contained (alle Verwaltungsinformationen vorhanden)
  - Software-Distribution und -Mitnutzung einfach („Container Registry“)
  - kleinere Images (nur eine Anwendung, keine Systemdienste)

# Anderer Leute Lösungen

# Anderer Leute Lösungen

- CyVerse / Agave (Web-basierte wiss. Analyse in der Cloud mit VMs)  
<http://www.cyverse.org/>, <https://agaveapi.co/>
- Adaptive Computing Viewpoint (Web-basierter Zugang zu trad. HPC)  
<http://www.adaptivecomputing.com/products/hpc-products/viewpoint/>
- Shifter (Software-Distribution mit Container Images für trad. HPC)  
<http://www.nersc.gov/research-and-development/user-defined-images/>
- Singularity (Container Runtime für trad. HPC)  
<http://singularity.lbl.gov/>
- Mikangelo (Geschwindigkeitsverbesserung von VMs für trad. HPC)  
<https://www.mikangelo-project.eu/>
- NICE EnginFrame (Web-basierte Visualisierung)  
<http://www.nice-software.com/products/enginframe/>

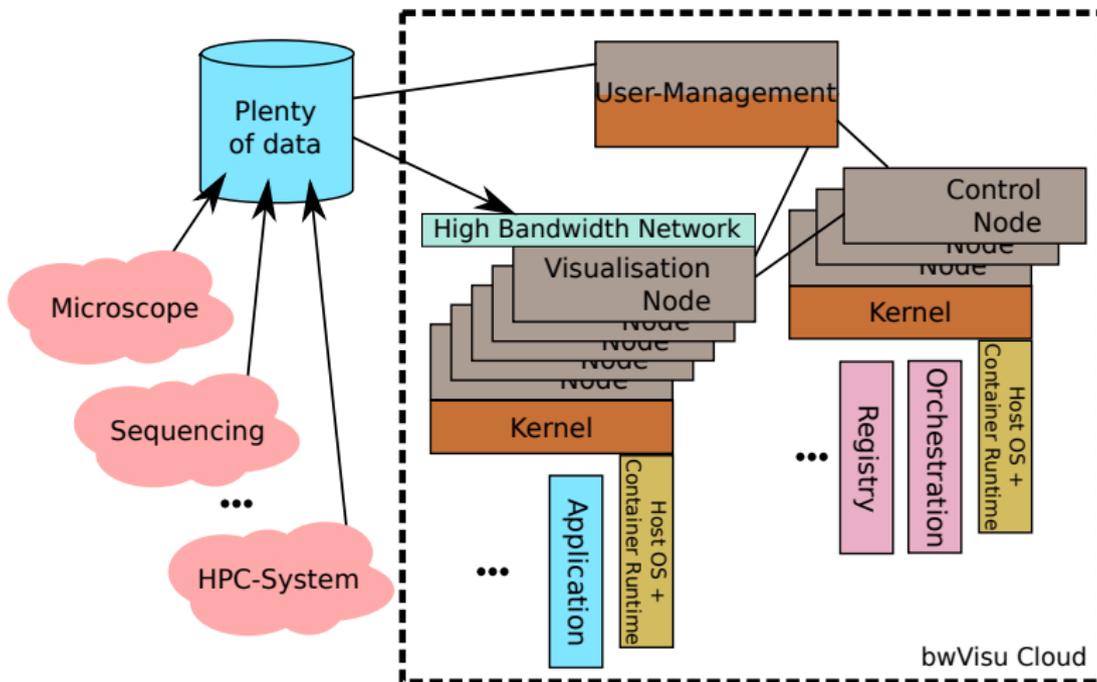


# Warum nicht bestehendes nutzen?

- Keines deckt alle Anforderungen ab
- Unser Fokus auf Visualisierung / interaktive Anwendungen
  - Alle genannten, außer EnginFrame, zielen auf Batch-Jobs ab
- Skalierbarkeit über lokalen Visualisierungs-Cluster hinaus schwierig

# bwVisu

# bwVisu Cloud



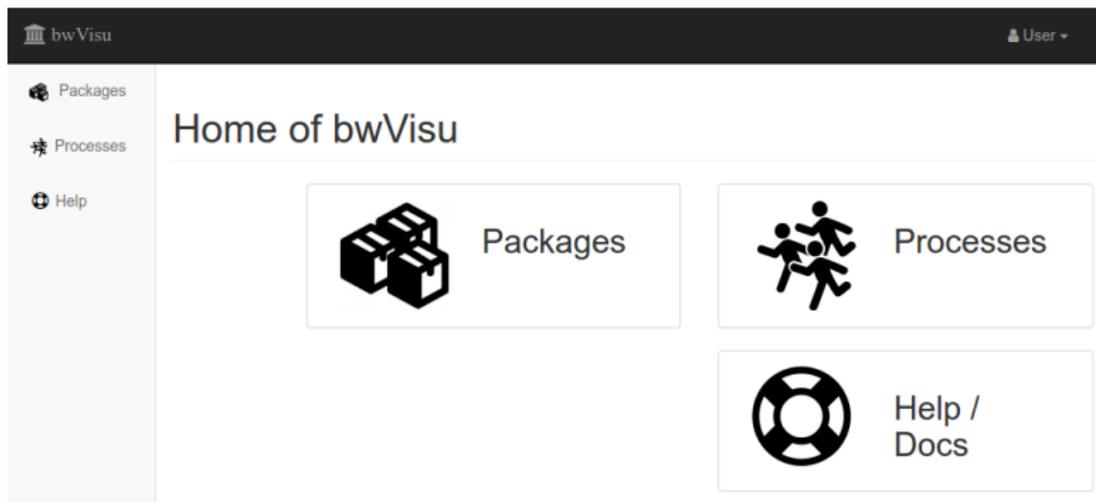


# bwVisu Cloud

- Software:
  - Orchestration Engine: Kubernetes (für Container-Betrieb ausgelegt)
  - Private Container Registry für lizenzpflichtige Software
  - „bwVisu (HTTP) API“ für Zugriff aus Web-Oberfläche
- Hardware:
  - Direkte (2x 40Gbps) Anbindung an lokalen Datenspeicher (SDS@hd)
  - Visualisierungsknoten: 2 CPU + 2 GPU, 0.25 TiB RAM, 1 TiB SSD
  - ca. 16 Knoten im Endausbau
- Nutzbar mittels bwIDM durch Landesnutzer



# bwVisu Web



Mock-Up: bwVisu Web home page



## Packages

### ParaView


[5.2.0](#)
[5.1.2](#)

#### Description

ParaView is an open-source, multi-platform data analysis and visualization application.

### HLRS Vistle


[git 20161117](#)
[git 20160804](#)

#### Description

A short description which contains only a few words.

Mock-Up: Package list



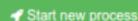
## ParaView 5.2.0



<b>Name</b>	ParaView
<b>Version</b>	5.2.0
<b>Updated</b>	2017-02-30
<b>Maintainer</b>	Dennis <dennis.schridde@uni-heidelberg.de>
<b>Homepage</b>	<a href="http://www.paraview.org/">http://www.paraview.org/</a>

### Processes:

- process #1
- mega Simulation XL
- my paraV
- more »

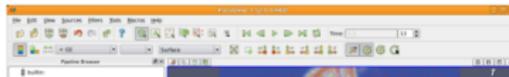
 Start new process

### Description

ParaView is an open-source, multi-platform data analysis and visualization application. ParaView users can quickly build visualizations of quantitative techniques. The data exploration can be done interactively in 3D or programmatically using ParaView's batch processing capabilities.

ParaView was developed to analyze extremely large datasets using distributed memory computing resources. It can be run on supercomputer size as well as on laptops for smaller data, has become an integral tool in many national laboratories, universities and industry, and has high performance computation.

### Screenshots



## Mock-Up: Package details



## Start ParaView - 5.2.0

Name

Containers  
(MPI-Processes)

Containersize

Comment

Mock-Up: Create a new process



# process #1 ParaView - 5.2.0

<b>Package</b>	ParaView - 5.2.0	<b>Stats</b>	
<b>Started</b>	2017-02-30 11:34	<b># CPUs (usage/limit)</b>	0.2 / 2
<b>Connect via</b>	129.206.10.25:58901	<b># GPUs (limit)</b>	4
		<b>RAM (usage/limit)</b>	5 MiB / 10 GiB
		<b>Costs</b>	2.35 "FairShare Poi

## How to connect

1. Start desktop application of ParaView
2. Goto Server-Settings and click **Add Server**



Mock-Up: Process details

# Zusammenfassung



# Zusammenfassung

- Schneller Datenzugriff dank guter Verbindung zu Datenspeicher
- Hohe vertikale Skalierbarkeit durch leistungsfähige Hardware
- Erhöhte horizontale Skalierbarkeit durch Virtualisierung
- Einfache Nutzung dank Container Images
- Nutzerfreundliche Web-Oberfläche



# Zusammenfassung

- Pilotdienst verfügbar
- Nutzeranfragen willkommen!
- Unterstützung beim Umsetzen Ihrer Use-Cases möglich
- → <http://bwvisu.de/>

Danke für Ihre Aufmerksamkeit!

# Fragen?

# Anhang



Dinesh Rajan u. a. „Converting a High Performance Application to an Elastic Cloud Application“. In: *Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science*. CLOUDCOM '11. Washington, DC, USA: IEEE Computer Society, 2011, S. 383–390. ISBN: 978-0-7695-4622-3. DOI: 10.1109/CloudCom.2011.58. URL: <https://doi.org/10.1109/CloudCom.2011.58>.



Aarthi Raveendran, Tekin Bicer und Gagan Agrawal. „A Framework for Elastic Execution of Existing MPI Programs“. In: *Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*. IPDPSW '11. Washington, DC, USA: IEEE Computer Society, 2011, S. 940–947. ISBN: 978-0-7695-4577-6. DOI: 10.1109/IPDPS.2011.240. URL: <http://dx.doi.org/10.1109/IPDPS.2011.240>.



Abhishek Verma u. a. „Large-scale cluster management at Google with Borg“. In: *Proceedings of the European Conference on Computer Systems (EuroSys)*. Bordeaux, France, 2015.



Ole Weidner u. a. „Rethinking High Performance Computing Platforms: Challenges, Opportunities and Recommendations“. In: *Proceedings of the ACM International Workshop on Data-Intensive Distributed Computing*. DIDC '16. Kyoto, Japan: ACM, 2016, S. 19–26. ISBN: 978-1-4503-4352-7. DOI: 10.1145/2912152.2912155. URL: <http://doi.acm.org/10.1145/2912152.2912155>.

