

MDDash: vše v jednom pro simulace molekulární dynamiky

Aleš Křenek, Filip Krása, et al.

Seminář výkonného počítání Metacentrum, 1.10.2026

MDDash – one stop shop for MD simulations

Filip Krása¹, Adrián Rošinec¹, Adam Ondrejka¹, and Aleš Křenek²

¹Masaryk university, ²CESNET

USER EXPERIENCE

Download from PDB or MDDb, provide file(s), or clone git repo

Initialize

Arbitrary MD simulation setup (solvate, add ions, minimize, equilibrate, ...)

Setup

Look for setup giving the best performance (ns/day)

Tune

Run the production simulation, watch progress

Simulate

Arbitrary analysis and visualization during and after simulation

Analyze

Prepare draft record to be published

Publish

IMPLEMENTATION

Data stored locally in **Kubernetes**, and synchronized to **S3**

Jupyter notebook(s), **Gromacs** available in sidecar container (light calls, immediate) or via K8s jobs (more resources)

RayTune to control short run experiments, try varying MPI/OMP cores, GPU assignment, ...

Create **Kubernetes** job, **allocate resources** according to tuning results

Jupyter notebook with suitable libraries (NGLview, mdtraj, ...); embedded **Mol*** for complex visualization

Call **Invenio** API, upload files including notebooks, extract metadata ...

MOTIVATION

1. MD Simulation setup is **complex** and frustrating, **difficult to reproduce**
2. Each simulation performs best with **specific resource configuration**
3. Spawning **long-running HPC jobs** is not trivial
4. Publishing results can be **error prone**

EOSC DATA COMMONS INTEGRATION

EOSC Data Commons project aims at **AI-assisted pairing** of datasets and computational workflows, spawning them on behalf of the users. MD is one of project **pilot use cases**, with MDDash as expected computation target.



MOTIVATION

1. MD Simulation setup is **complex** and frustrating, **difficult to reproduce**
2. Each simulation performs best with **specific resource configuration**
3. Spawning **long-running HPC jobs** is not trivial
4. Publishing results can be **error prone**



USER EXPERIENCE

Download from PDB or MDDB,
provide file(s), or clone git
repo

Initializ

Arbitrary MD simulation setup
(solvate, add ions, minimize,
equilibrate, ...)

provide file(s), or clone git repo

Initialize

Arbitrary MD simulation setup
(solvate, add ions, minimize, equilibrate, ...)

Setup

Look for setup giving the **best performance** (ns/day)

Tune

(solvate, add ions, minimize, equilibrate, ...)

Look for setup giving the **best performance** (ns/day)

Tune

Run the **production simulation**,
watch progress

Look for setup giving the **best**
performance (ns/day)

Tune

Run the **production simulation**,
watch progress

Simulate

Arbitrary **analysis** and **visual-
ization** during and after simu-
lation

Analyze

from the **production simulation**,
watch progress

Arbitrary **analysis** and **visualization** during and after simulation

Analyze

Prepare **draft record** to be
published

ization during and after simulation

Analyze

Prepare draft record to be published

Publish

MOTIVATION

EOSC

Živé demo

MDDash – one stop shop for MD simulations

Filip Krása¹, Adrián Rošinec¹, Adam Ondrejka¹, and Aleš Křenek²

¹Masaryk university, ²CESNET

USER EXPERIENCE

Download from PDB or MDDb, provide file(s), or clone git repo

Initialize

Arbitrary MD simulation setup (solvate, add ions, minimize, equilibrate, ...)

Setup

Look for setup giving the best performance (ns/day)

Tune

Run the production simulation, watch progress

Simulate

Arbitrary analysis and visualization during and after simulation

Analyze

Prepare draft record to be published

Publish

IMPLEMENTATION

Data stored locally in Kubernetes, and synchronized to S3

Jupyter notebook(s), Gromacs available in sidecar container (light calls, immediate) or via K8s jobs (more resources)

RayTune to control short run experiments, try varying MPI/OMP cores, GPU assignment, ...

Create Kubernetes job, allocate resources according to tuning results

Jupyter notebook with suitable libraries (NGLview, mdtraj, ...); embedded Mol* for complex visualization

Call Invenio API, upload files including notebooks, extract metadata ...

MOTIVATION

1. MD Simulation setup is complex and frustrating, difficult to reproduce
2. Each simulation performs best with specific resource configuration
3. Spawning long-running HPC jobs is not trivial
4. Publishing results can be error prone

EOSC DATA COMMONS INTEGRATION

EOSC Data Commons project aims at AI-assisted pairing of datasets and computational workflows, spawning them on behalf of the users. MD is one of project pilot use cases, with MDDash as expected computation target.



IMPLEMENTATION

Initialize

Data stored locally in **Kubernetes**, and synchronized to **S3**

Setup

Jupyter notebook(s), **Gromacs** available in sidecar container (light calls, immediate) or via

netes, and synchronized to **S3**

Setup

Jupyter notebook(s), **Gromacs** available in sidecar container (light calls, immediate) or via K8s jobs (more resources)

RayTune to control short run experiments, try varying MPI/OMP cores, GPU assign-

Setup

(light calls, immediate) or via
K8s jobs (more resources)

Tune

RayTune to control short
run experiments, try varying
MPI/OMP cores, GPU assign-
ment, ...

Simulate

Create Kubernetes job, **allo-
cate resources** according to
tuning results

run experiments, try varying
MPI/OMP cores, GPU assign-
ment, ...

Simulate

Create Kubernetes job, **allo-
cate resources** according to
tuning results

Jupyter notebook with suitable
libraries (NGLview, mdtraj, ...);
embedded **Mol*** for complex

Simulate

cate resources according to tuning results

Analyze

Jupyter notebook with suitable libraries (NGLview, mdtraj, ...); embedded **Mol*** for complex visualization

Publish

Call **Invenio** API, upload files including notebooks, extract metadata

libraries (NGLEview, Matplotlib, ...),
embedded **Mol*** for complex
visualization

Publish

Call **Invenio** API, upload files
including notebooks, extract
metadata ...

EOSC DATA COMMONS INTEGRATION

MOTIVATION

1. MD Simulation setup is **complex** and frustrating, **difficult to reproduce**
2. Each simulation performs best with **specific resource configuration**
3. Spawning **long-running HPC jobs** is not trivial
4. Publishing results can be **error prone**



Chci začít

- Prozatím experimentální provoz
 - přístup na požádání
 - časté změny
 - možná nestabilita, bez garance zachování dat
- <https://mddash.dyn.cloud.e-infra.cz/>