



Advancements of the Fusion Digital Twin: new architecture, structure and services

C. Sommariva¹, F. Cabot², D. Fransos³, S. Kerboua-Benlarbi¹, C. Angerer³, C. Favreau³, F. Yang³, H. Elaian³, M. Toussaint¹, A. Fedrigucci⁴, A. Pau¹, M. Cai³, P. Blachard¹, S. Duthaler³, H. Reimerdes¹, O. Sauter¹, E. Tonello¹, C. Yildiz¹, J. Zielinski³

¹EPFL-SPC, CH-1015, Lausanne, Switzerland

²EPFL-SCITAS, CH-1015, Lausanne, Switzerland

³NVIDIA Corporation, 2788 San Tomas Expressway, Santa Clara, California, 95051, USA

⁴EPFL-THEOS-MARVEL, CH-1015, Lausanne, Switzerland



Swiss
Plasma
Center



This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Fusion Digital Twin Proof-of-Concept



Fusion Digital Twin Proof-of-Concept: digital copy of multiple TCV divertor configurations

⇒ Demonstrate integrated physics-engineering computing, surrogate models, advanced rendering & database handling

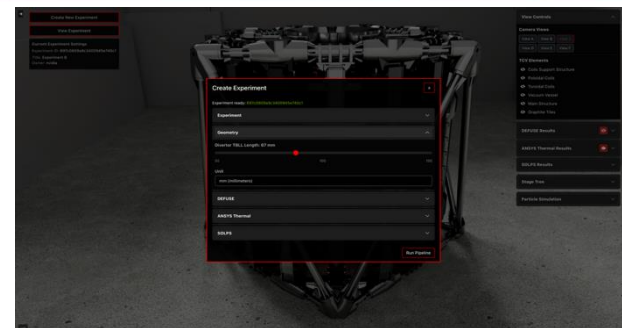
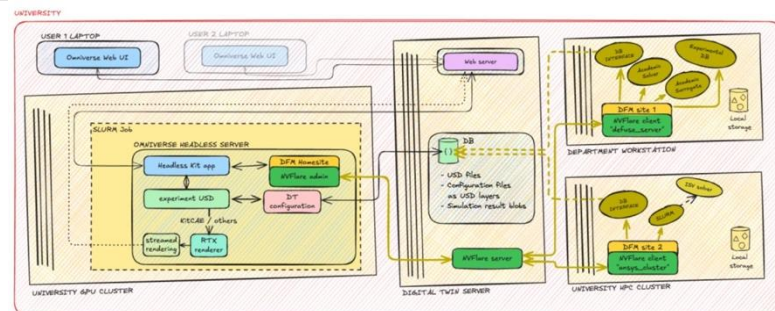
Technology of choice: NVIDIA Omniverse Ecosystem

⇒ Data Federation Mesh (DFM): orchestrate execution of programmable pipeline & data transfer, enables distributed pipeline execution on different hardware

⇒ Omniverse KIT: SDK for building omniverse applications, advanced FDT GUI and photorealistic rendering (RTX)

⇒ OpenUSD based database (3D scene description format)

⇒ Proven the integrated edge plasma-thermal modeling of TCV divertor equipped w GUI, rendering & database



Remote rendering on HPC cluster



Remote photorealistic rendering on GPU HPC cluster under testing

⇒ GUI executed on VM while rendering executed as slurm job on GPU HPC cluster

- User connects to web-service on VM via web-browser
- Web-service submit rendering job on GPU

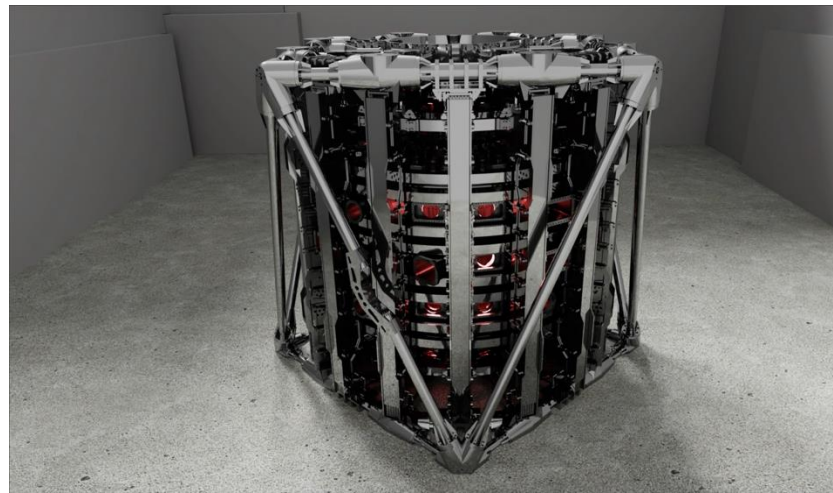
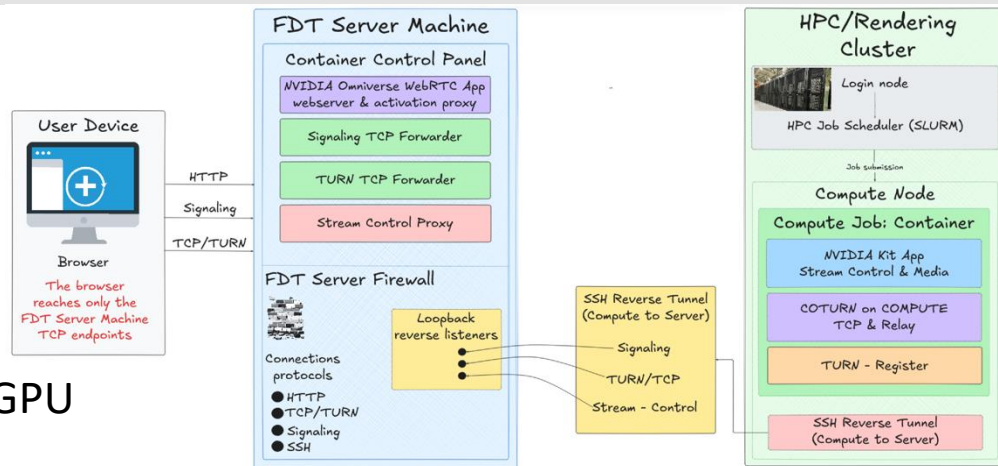
⇒ GPU cluster execute the rendering job

- Requires cluster equipped w RTX HW
- Neuro-shader tech. beneficial

⇒ Rendering data returned and displayed in GUI

Advantages

- ⇒ No specific HW required by users
- ⇒ No specific installation required by users
- ⇒ Multiple users can connect and use FDT



Service Layout: structure, testing, ...



Name	Last commit	Name	Last commit	Name	Last commit
config	renamed services environment variab...	..		..	
datasets/fdt_software	Integrated softs service within the new...	ansys	bugfixes system_tools, change name of the...	__init__.py	restructured slurm service tree
doc/fdt_software	Integrated softs service within the new...	defuse	updated dependencies for building and inst...	slurm_config.py	bugfixed fdt_services. in import
fdt_services	renamed services environment variab...	matlab	updated dependencies for building and inst...	slurm_job.py	bugfixed fdt_services. in import
models/fdt_software/direct_train...	models: re-dump production jobs sh...	slurm	bugfixed fdt_services. in import	slurm_job_status.py	restructured slurm service tree
scripts	reporting py: switch to SQLPredictor ...	soifs	updated dependencies for building and inst...	slurm_service.py	bugfixed fdt_services. in import
tests	renamed services environment variab...	tools	bugfixes system_tools, change name of the...	slurm_service_execute.py	bugfixed fdt_services. in import
gitattributes	setup git attributes and ignore, added ...				
gitignore	removed .env from tracked file				
gitlab-ci.yml	gitlab-ci.yml: fix artifacts path -> pat...				
LICENSE	working pyproject.toml for correct pac...				
README.md	Modify bash script and remove pyproj...				
env.example	small bugfix of the folder naming in ...				
pyproject.toml	small bugfix for avoiding building glob...				
setup_python_environment.sh	removed debug scripts				

FDT services are in refactoring

⇒ Services coded as an as-independent-as-possible python package

- ⇒ Re-use in different FDT config.
- ⇒ Re-use outside FDT if needed
- ⇒ Install only site-required services

⇒ Currently, services are code-only

- ⇒ Configuration & pre-set in JSON
- ⇒ Data and models are shipped outside the service package
- ⇒ Liking via JSON dictionary

Please log in

Username

Password

Log in

Files and folders in /var/www/html/fdt_services_data/data

Remove selected

☐	Name	Type	Size	Last Modified	Action
☐	ansys_test.zip	File	11.66 MB	2026-08-07 13:20:05	Remove

Total items: 1

Log out

Upload file

Upload file

Unit testing for services currently under develop

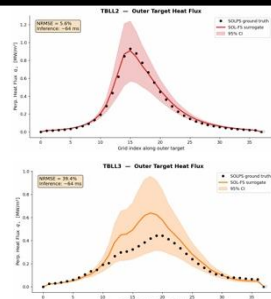
- ⇒ Unit testing based on pytest engine
- ⇒ Clear separation between fixtures and methods
 - ⇒ Same fixture reused by multiple methods
- ⇒ Prototype of webservice for storing FDT and test data developed (in php) and running

FDT services development status



SOL-FS service

- SOLPS-ITER surrogate model
- Heat and particle fluxes
- Refactoring completed!
- DB under construct.
- workflow for DB gen in dev

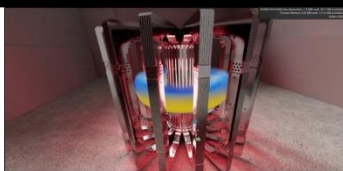


SLURM service

- Prepare-inputs, submit, retrieve data, garbage collect of SLURM jobs
- Service restructuring in as python package completed!
- Integration of refactored package in FDT on-going

DEFUSE srv

- Load exp. plasma data
- I/O FDT data DB (in IMAS)
- Python package ready (UT req)
- Integration in FDT TBS
- Ext. for executing data manip. & analysis pipeline in discuss.

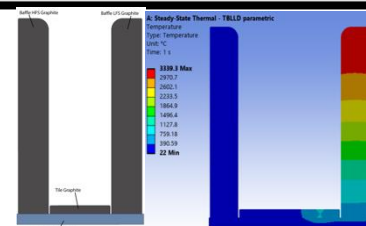


MatLab service

- Start/Stop Mat shared session
- Insert, retrieve data and execute MatLab cmd from Py
- Python package ready
- UT almost completed
- Integration in FDT TBS

Ansyz serv.

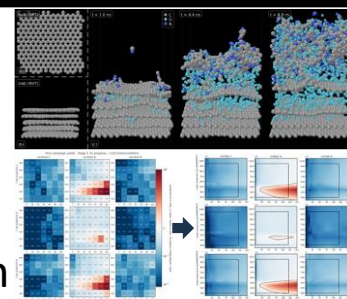
- Meshing & thermal simulation
- Refactoring almost completed
- Multi-parameters CAD in dev.
- Integration new srv in FDT TBS
- Ext to 3D mechanical sim. TBS



Carbon

Sputtering

- Infers yield, distrib of CS from MD sim
- Sim DB constructed & interpolator developed
- Service implementation TBS



Conclusions and Future Work



Conclusions

- FDT (pre-)production configuration & computational resources defined
 - Documentation in drafting
 - Installation in progress
- Prototype of streaming rendering developed and currently under testing
- FDT services in advanced refactoring
 - Services defined as python package
 - Configuration stored in JSON dict.
 - Data stored outside the packaging
 - Unit testing in implementation
- SOLPS-ITER DB generation workflow in refinement & C sputtering database ready

Future work

- Finalize (pre-)production installation of FDT
- Update FDT integrating streaming rendering
- Finalize documentation of both orchestrator and streaming rendering technology
- Finalize services refactoring and unit testing
- Update FDT for exploiting refactored services
- Finalize SOLPS-ITER DB generation workflow
- Implement Carbon Sputtering service
- Extend Ansys service calculation to 3D multi-parameter thermo-mechanical simulations
- Enrich SOLFS with radiative source inference
- Project on robot control system dev. started
- Develop B-line, particle and ray tracing serv.