Hypernet.

Towards Scalable Parametric/ Stochastic 2D Hydrodynamic Modeling via Process Containerization

Todd Chapman, Ph.D.

CTO, Hypernet Labs

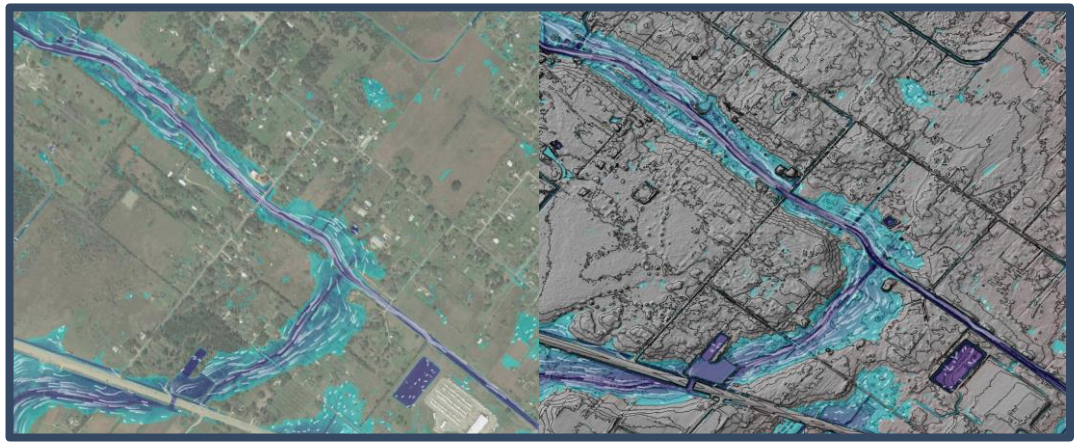
Redwood City, CA

MOTIVATION

Hypernet.

2D Hydrodynamic modeling is computationally intensive

- **1M+ grid cells** is commonplace
- A single project can require dozens of scenarios, each taking **multiple days to complete**
- Uncertainty quantification via Monte Carlo is likely to become **mandatory** in the near future
    - ~10,000 realizations to reach statistical convergence
    - Practitioners often resort to model truncation or simplification to achieve feasibility



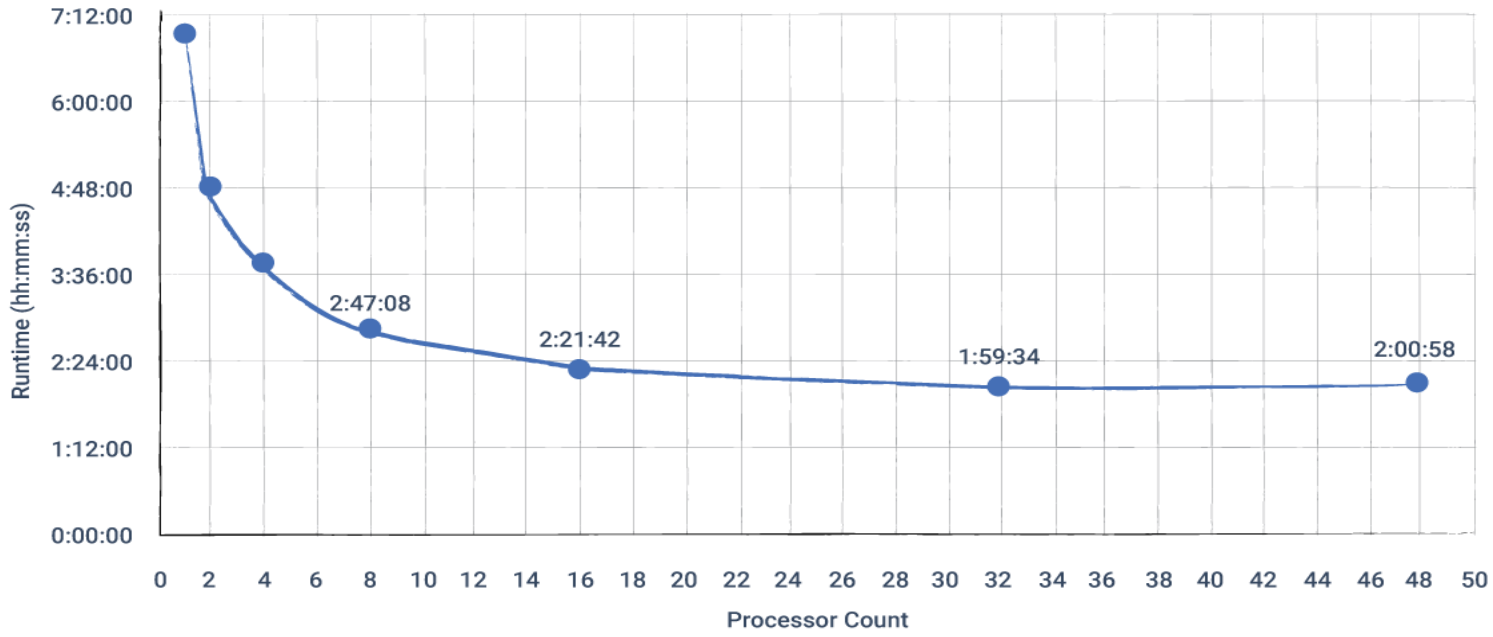
Two screenshots showing the split flow nature of a stream and how runoff is diverted based on the underlying terrain. Runtime: 24+ hours. Photo Credit: Luis Partida, RPS

CURRENT STATE OF THE ART

Hypernet.

Many physics kernels used in the hydrodynamic modeling process were designed for desktop/personal computing environments (SWMM, HEC-RAS, SRH2D, etc.)

- **Local resources are limited** in core count and memory amount
- **Moore's Law** can no longer be leveraged for passive performance boosts [Theis & Wong]
- **Limited scalability** for thread-based, shared-memory numerical algorithms



APPROACHES TO ACCELERATE LEGACY SIMULATION ENGINES

Intrusive

Hypernet.

- **Model Order Reduction**
  - Works very well for **linear dynamics**, **10,000x speedup** has been demonstrated in various applications (Acoustics, VLSI Integrated Circuit Design, Linearized Solid Mechanics) [Farhat, *et. al.*]
  - **Data-driven**: leverage database of historical results to algebraically reduce the number of degrees of freedom
  - **Difficult to achieve accuracy and speedups** for hyperbolic, nonlinear models. [Chapman, *et. al.*]

APPROACHES TO ACCELERATE LEGACY SIMULATION ENGINES

Intrusive

Hypernet.

- **Distributed Memory Parallelization**
  - Requires **re-writing large portions** of the codebase to handle distributed data structures (particularly for implicit time-stepping) with MPI library [Gropp, *et. al.*]
  - Likely also requires the target OS to be a **linux variant** to get full benefit
  - Past verification & validation is **no longer applicable** to the new codebase
  - Requires a **cluster with high-speed interconnects**

APPROACHES TO ACCELERATE LEGACY SIMULATION ENGINES

Intrusive

Hypernet.

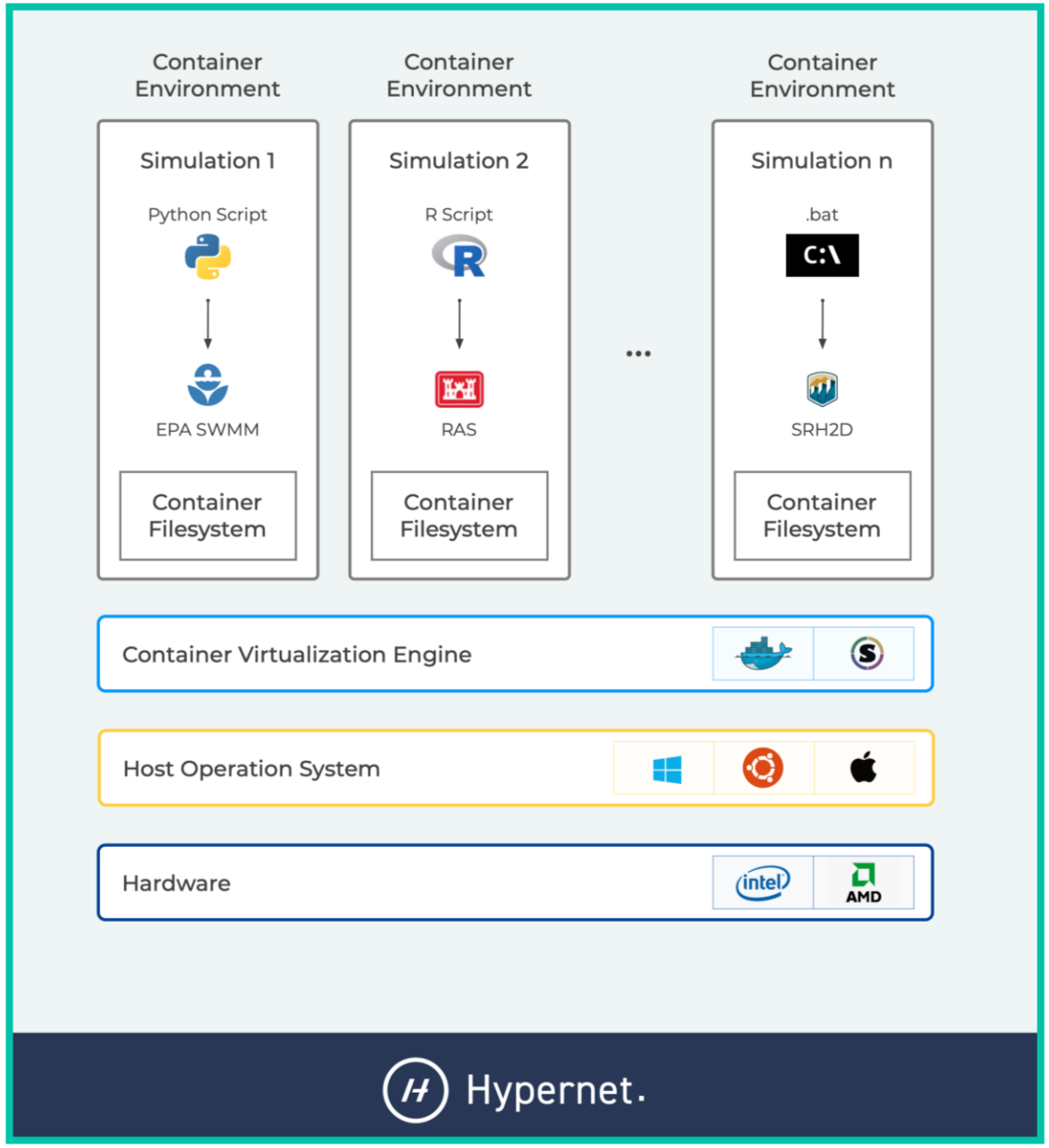
- **GPU Acceleration**
  - Capable of providing **massive speedups** without needing access to a supercomputer
  - As with distributed memory parallelism, requires **re-writing** of time-stepping algorithms
  - **Instabilities** associated with numerical round-off [Jézéquel, *et. al.*]
  - Performance is highly dependent on the GPU architecture making code **less portable**

APPROACHES TO ACCELERATE LEGACY SIMULATION ENGINES

Non-Intrusive

Hypernet.

- **Resource Scaling**
  - Temporarily acquire a large amount of high (or low) reliability resources and run all required scenarios **simultaneously**
  - Uses **available simulation engines** with no modification
  - **Challenge:** environment consistency, deployment pipeline, results pipeline



Hypernet.

CONTAINERIZATION AS A SCALING MECHANISM

Containerization provides portability and a **uniform interface** to any simulation stack

CONTAINERIZATION AS A SCALING MECHANISM

Hypernet.

- **Pros:**
  - Package data, dependencies, and application in **one lightweight package**
    - i.e., package SWMM5 with python interpreter, associated modules and scripts, and your input files and know you'll get the same results every time you run the stack
  - Provides a **uniform interface** to all simulation engines
    - Simulator complexity is abstracted away by container runtime interface
  - **Easy to manage** versions and distribute updates
    - Many options for container registries
    - Most major container technologies implement efficient image caching

CONTAINERIZATION AS A SCALING MECHANISM

Hypernet.

- **Cons:**
  - **Steep learning curve** for virtualization of new applications
    - Dockerfiles can sometimes be difficult to write
  - Model management doesn’t come **out-of-the-box**
    - You still need a pipeline for the deployment and monitoring of simulations and retrieval of output files

LET’S SEE WHAT IT WOULD LOOK LIKE FOR SWMM5

Hypernet.

- Check out the **Hypernet Labs github** repository for a full example
    - Packages a Python interpreter and runtime script alongside all major releases of the SWMM5 and Open-SWMM engines
    - You can try it out yourself and make your own customizations
- Let's see how this would fit into a **deployment pipeline**

CITATIONS

Hypernet.

Theis, T. N., & Wong, H. S. P. (2017). The end of moore's law: A new beginning for information technology. *Computing in Science & Engineering*, *19*(2), 41-50.

Gropp, W., Gropp, W. D., Lusk, E., Skjellum, A., & Lusk, A. D. F. E. E. (1999). *Using MPI: portable parallel programming with the message-passing interface* (Vol. 1). MIT press.

Farhat, C., Avery, P., Chapman, T., & Cortial, J. (2014). Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, *98*(9), 625-662.

Chapman, T., Avery, P., Collins, P., & Farhat, C. (2017). Accelerated mesh sampling for the hyper reduction of nonlinear computational models. *International Journal for Numerical Methods in Engineering*, *109*(12), 1623-1654.

Jézéquel, F., Lamotte, J. L., & Saïd, I. (2015, September). Estimation of numerical reproducibility on CPU and GPU. In *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 675-680). IEEE.