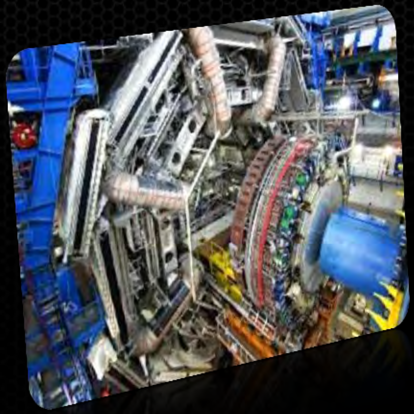
In-Situ Computing: Supercomputing at the Source

Peter Messmer

A typical Instrument

Raw Resource



Experiment

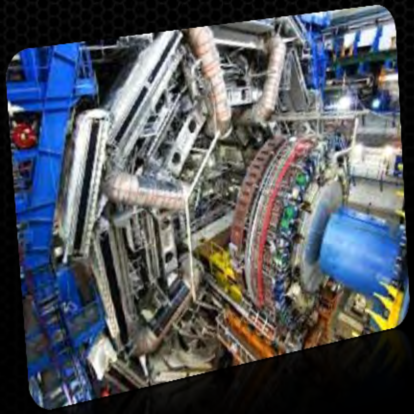
Trigger

Offline processing

Off-site processing

A typical Instrument

Raw Resource



Experiment

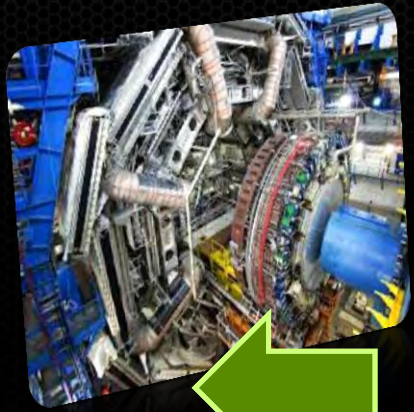
Trigger

Offline processing

Off-site processing

A typical Instrument

Raw Resource



Experiment

Trigger

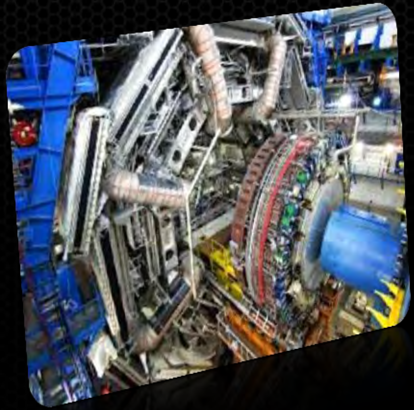
Offline processing

Off-site processing

- No OS
- Priority streams
- GPU Direct RDMA

Supercomputer as an Instrument

Raw Resource

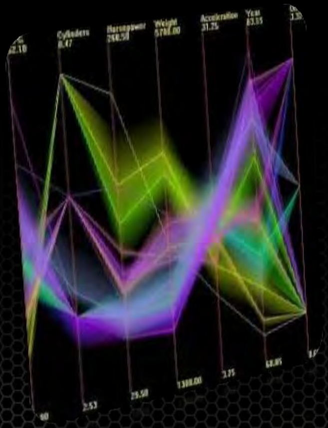


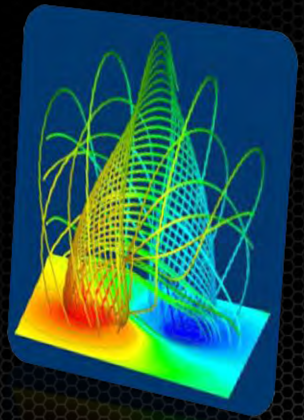
Experiment

Trigger

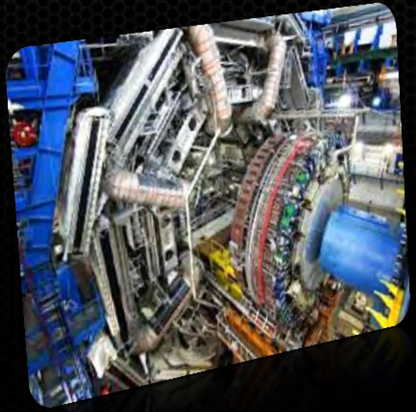
Offline processing

Off-site processing



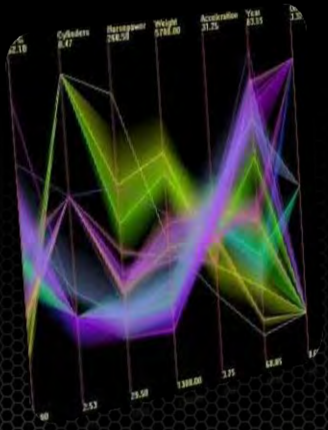


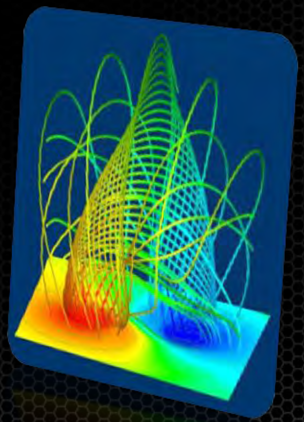
Supercomputer as an Instrument



Raw Resource Experiment Trigger Offline processing Off-site processing

?





HEP trigger design

- Objectives
  - Select interesting data
  - Pre-process suitable for higher level triggers
  - Non-intrusive, non-disruptive
- Challenge: selective enough, but not too restrictive
  - Not only look for science that’s already known
- Trends: More flexibility
  - Discover the unexpected

HEP trigger design

- Objectives
  - Select interesting data
  - Pre-process suitable for higher level triggers
  - Non-intrusive, non-disruptive

Same applies to HPC "triggers"

- Challenge: selectiv… not too restrictive
  - Not only look for sc…ence that's already known

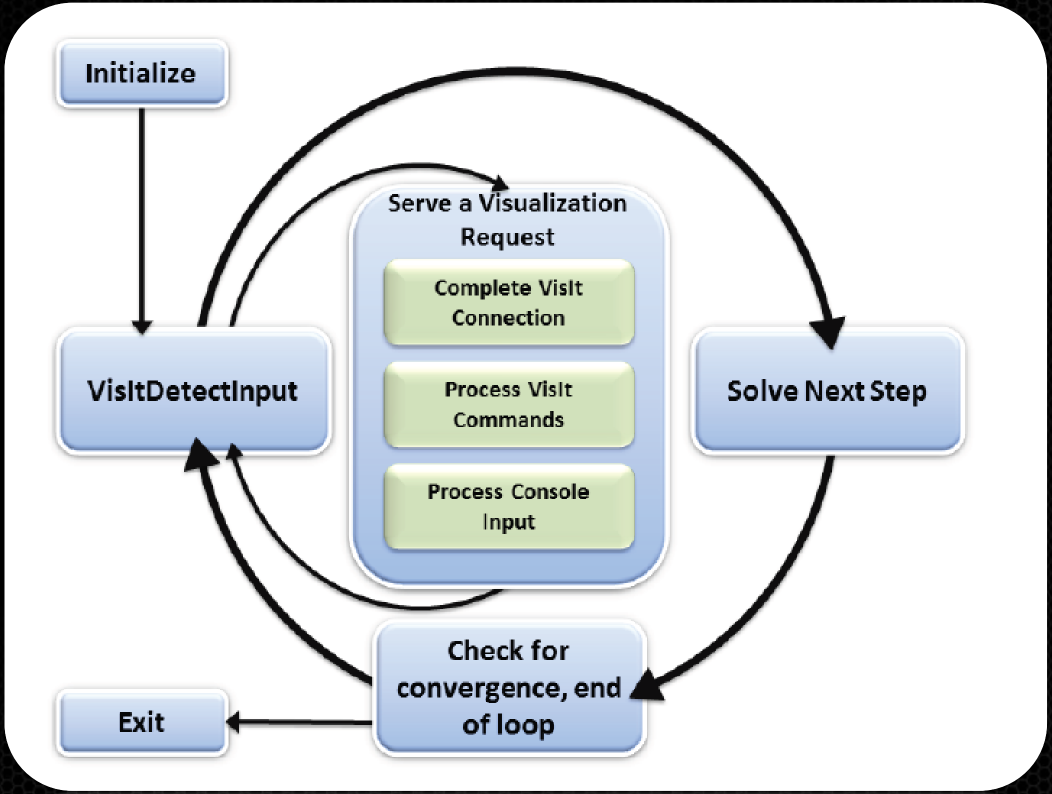
- Trends: More flexibility
  - Discover the unexpected

What are the triggers in HPC?

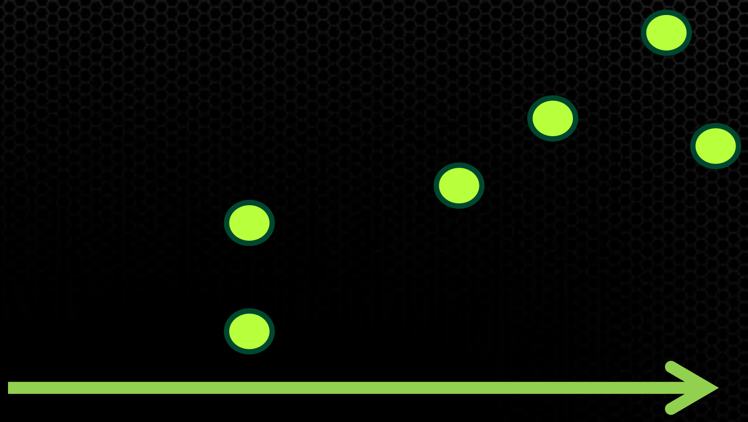
- Application specific reductions/analysis
  - E.g. particles -> densities
  - Rigid
- In-situ visualization/analysis
  - Supported by common viz tools, support for GPUs
  - Flexibility
- Find commonality among the triggers
- Incorporate technologies from computer vision, machine learning, ..

In-situ visualization

- Visit: libsim
- Paraview: Catalyst
- Reduce/analyze on the fly
- GPU accelerated
- Objective of SDAV SciDAC-3



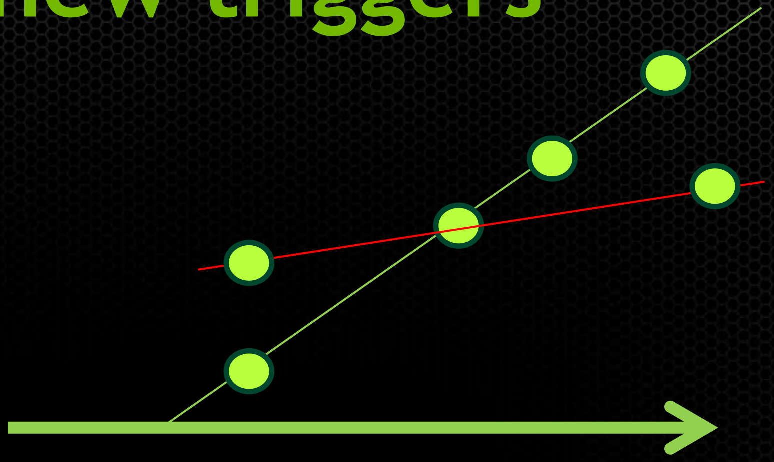
Compute power enables new triggers



- Detector measures “hits”
- Find points that form a track
  - Assumption: Track originates from central axis
  - Often the case, but what if not?

Compute power enables new triggers

- Detector measures “hits”
- Find points that form a track
    - Assumption: Track originates from central axis
    - Often the case, but what if not?



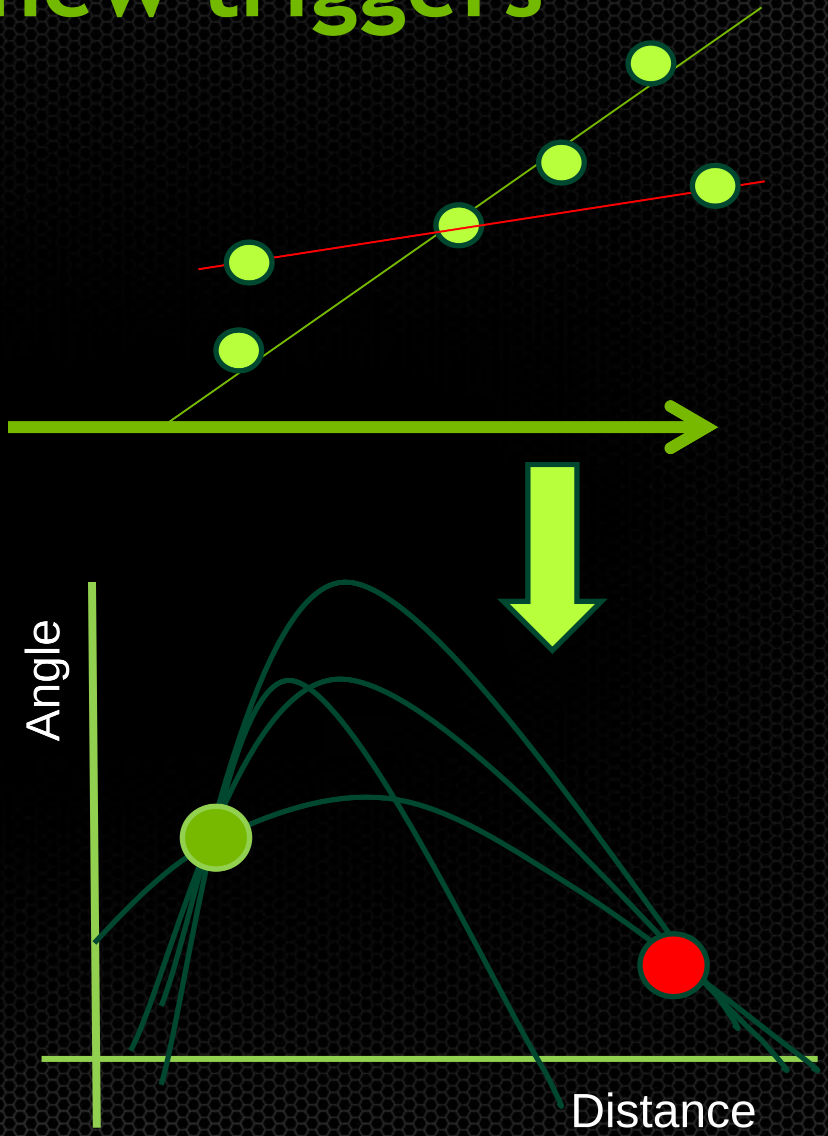
Compute power enables new triggers

- Detector measures “hits”
- Find points that form a track
  - Assumption: Track originates from central axis
  - Often the case, but what if not?

- “Hough”-transform hits into curves
  - Statistics on most likely tracks

- Detect “unexpected” events

=> Computer Vision meets HPC



GPUs are Going Beyond Scientific & Technical Computing

Big Data Analytics

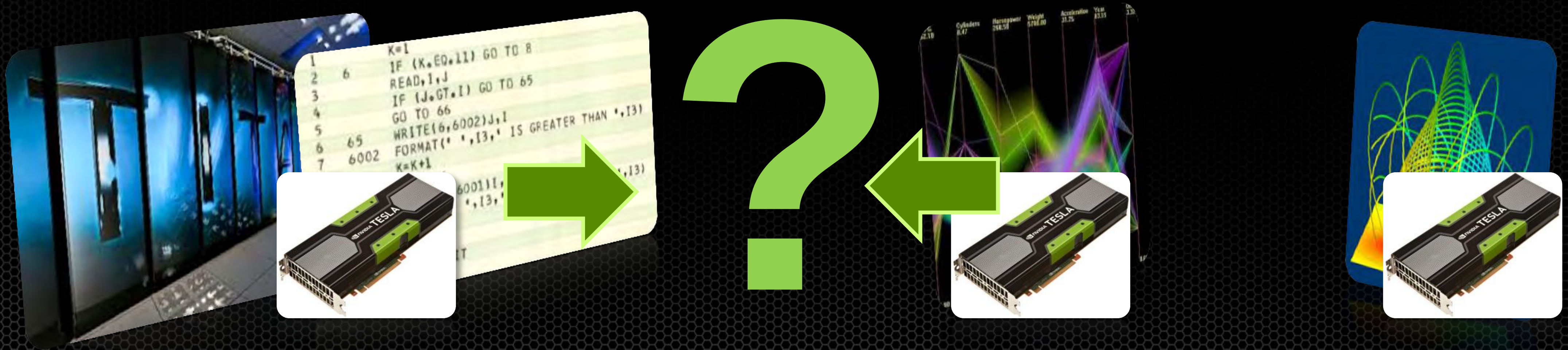
Machine Learning

Computer Vision

Analyzing Twitter	salesforce
Searching Audio	SHAZAM
Visual Shopping	CORTEXICA vision systems; ebay
Real-time Video Delivery	ELEMENTAL; ESPN

Supercomputer as an Instrument

Raw Resource | Experiment | Trigger | Offline processing | Off-site processing



Overarching Goals for Tesla

Power
Efficiency

Ease of
Programming
And Portability

Application
Space
Coverage

3 WAYS TO ACCELERATE APPLICATIONS

Applications

Libraries	OpenACC Directives	Programming Languages
"Drop-in" Acceleration	Easily Accelerate Applications	Maximum Flexibility

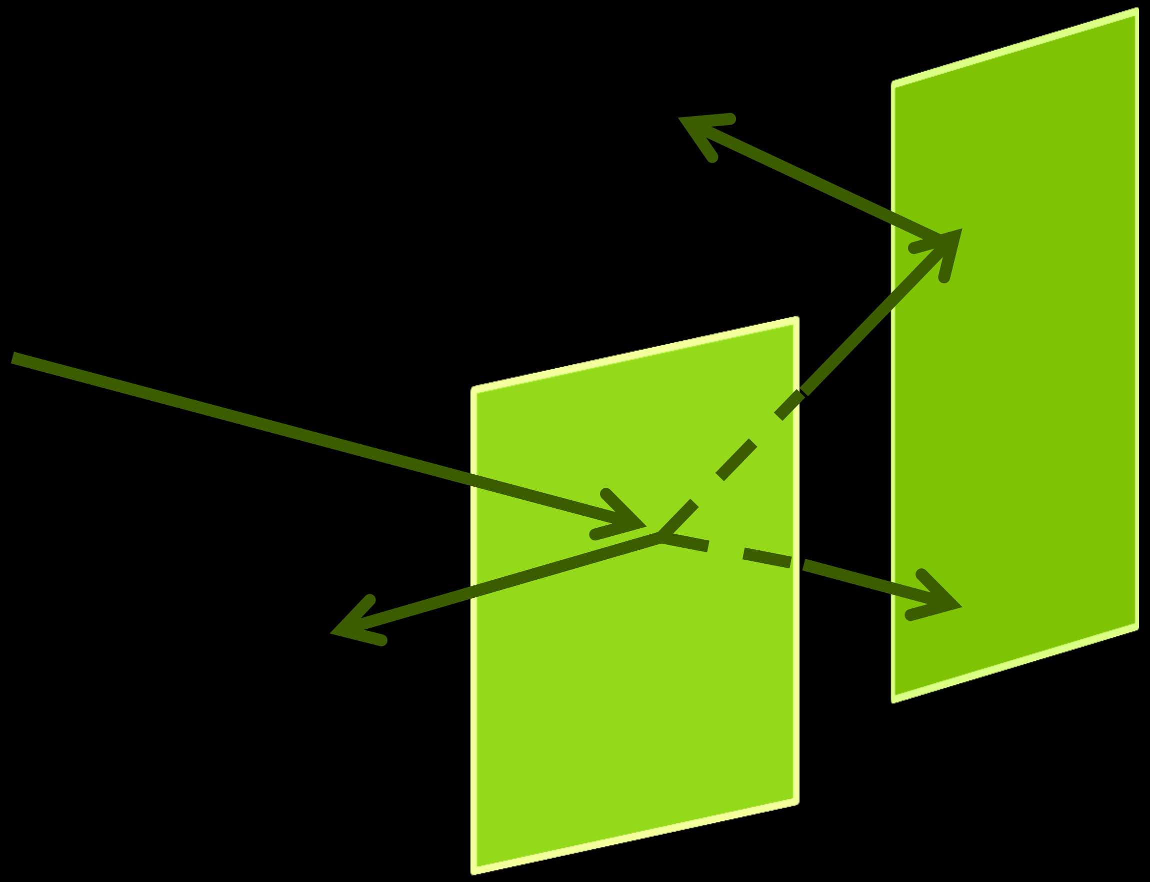
3 WAYS TO ACCELERATE APPLICATIONS

"Drop-in" Acceleration

Easily Accelerate Applications

Maximum Flexibility

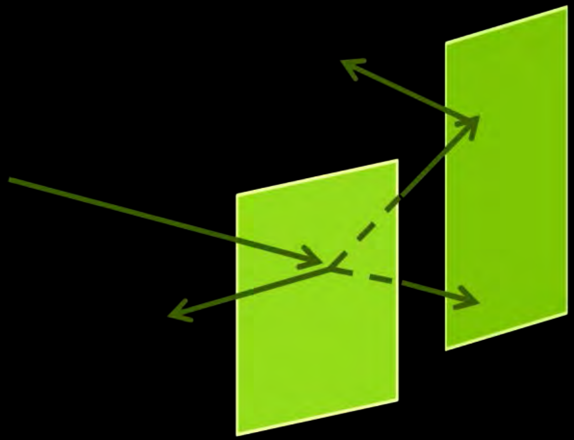
IF YOUR APPLICATION LOOKS LIKE THIS..



.. YOU MIGHT BE INTERESTED IN OPTIX

- Ray-tracing framework
  - Build your own RT application

- Generic Ray-Geometry interaction
  - Rays with arbitrary payloads

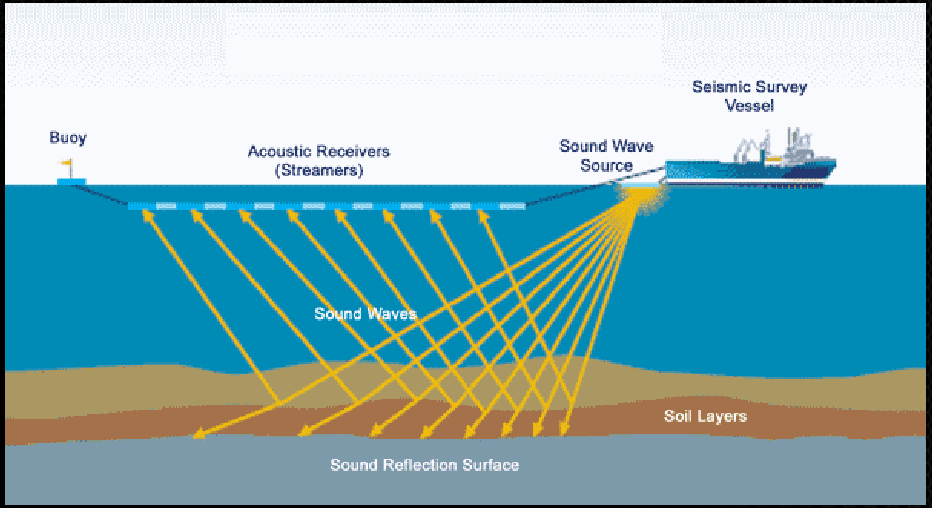


- Multi-GPU support

OptiX can Simplify Scene Queries

- Seismic wave simulation code
  - Unstructured mesh simulation
  - Needs to know location of seismogram on the mesh
  - Brute force algorithm possible

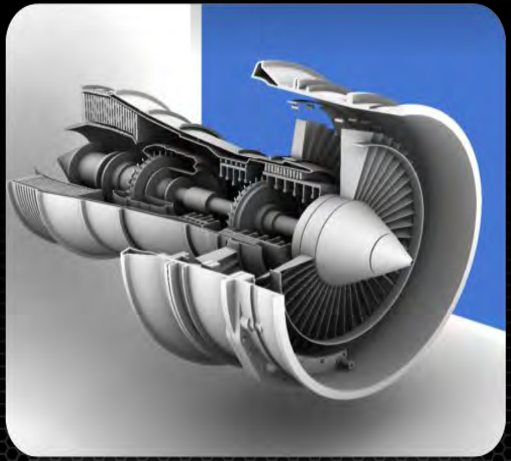
- Now want this time dependent, and GPU accelerated
  - Simple N^2 algorithm doesn't work
  - Maybe something tree based?



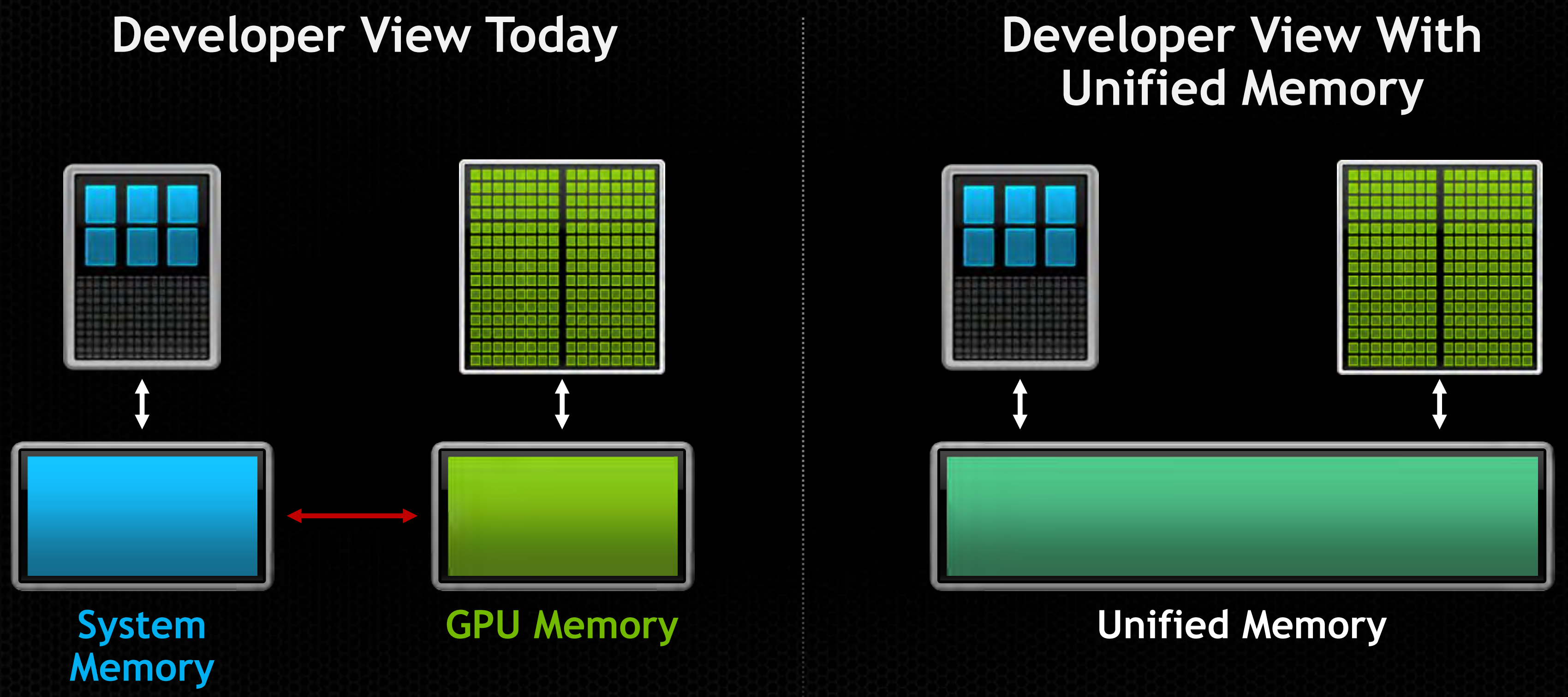
=> Scientists don't want to spend their time writing scene query codes

OptiX Prime: Low-Level Ray Tracing API

- OptiX simplifies implementation of RT apps
  - Manages memory, data transfers etc
- Sometimes an overkill for simple scene queries
  - E.g. just need visibility of triangulated geometries
- OptiX Prime: Low-Level Tracing API



Unified Memory
Dramatically Lower Developer Effort



Super Simplified Memory Management Code

CPU Code

```
void sortfile(FILE *fp, int N) {
  char *data;
  data = (char *)malloc(N);

  fread(data, 1, N, fp);

  qsort(data, N, 1, compare);


  use_data(data);

  free(data);
}
```

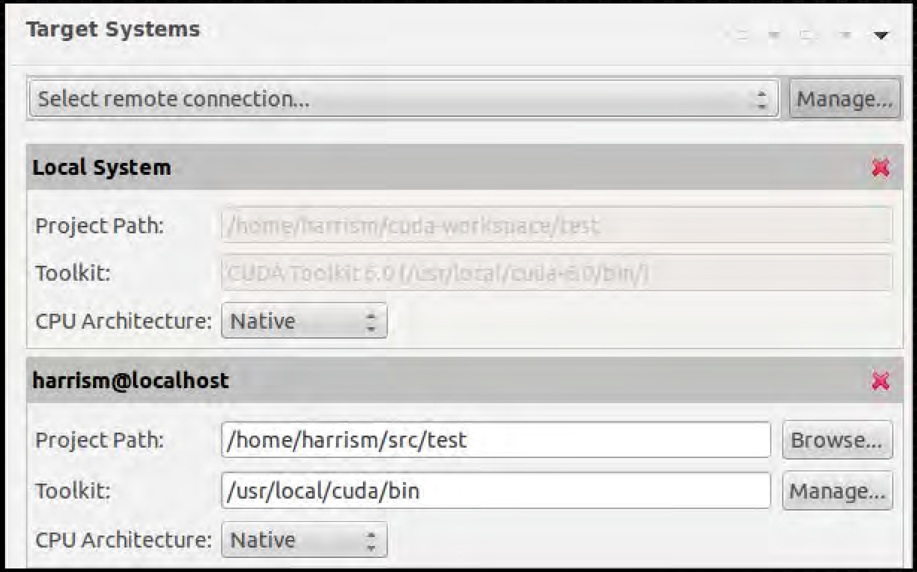
CUDA 6 Code with Unified Memory

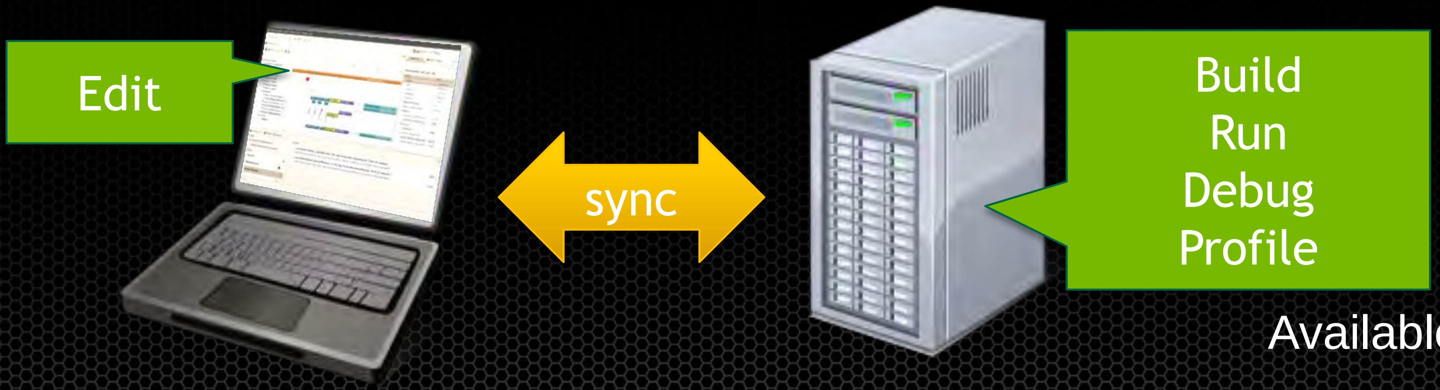
```
void sortfile(FILE *fp, int N) {
  char *data;
  cudaMallocManaged(&data, N);

  fread(data, 1, N, fp);

  qsort<<<...>>>(data,N,1,compare);
  cudaDeviceSynchronize();

  use_data(data);

  cudaFree(data);
}
```

Available in CUDA 6

Remote Development with Nsight Eclipse Edition

- Local IDE, remote application
  - Edit locally, build & run remotely
  - Automatic sync via ssh
  - Cross-compilation to ARM
- Full debugging & profiling via remote connection

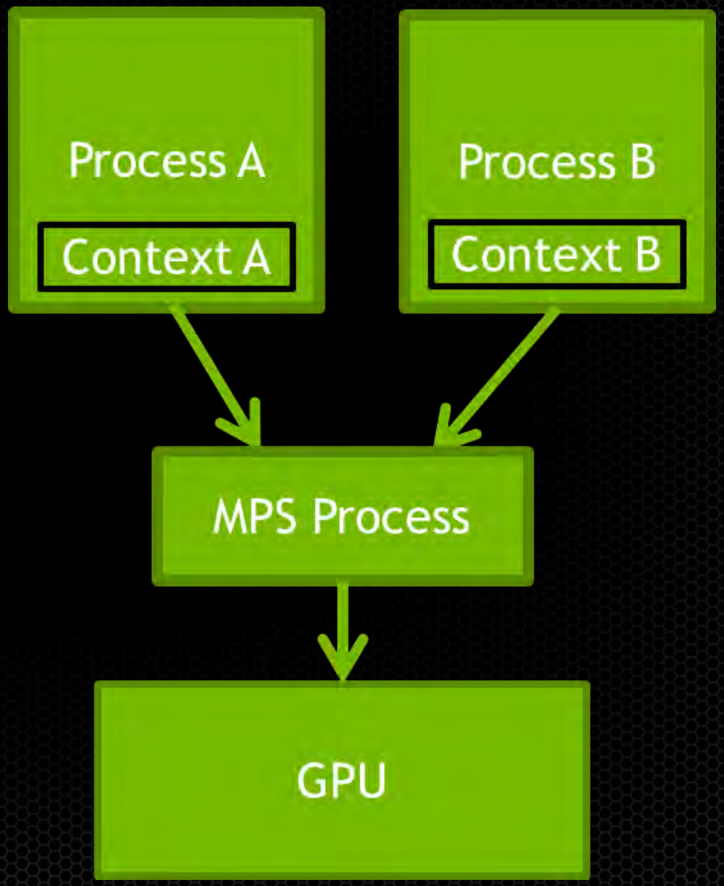




Available in CUDA 6

Multi-Process Server MPS

- Strong scaling => GPU utilization can
  become limited
- MPS: Share GPU among multiple MPI-ranks
- Not limited to MPI codes
  - Share GPU for compute and data reduction



Summary/Conclusions

- Supercomputers very close to Instruments
  - Main discrepancy in “triggering”
- GPU acceleration enables advanced triggering
  - In-situ visualization, computer vision, machine learning..
- Middleware layers can help
  - Look outside of “conventional” HPC
  - Attract wider crowd of developers
  - Longer term: Design algorithms with middleware in mind
- GPU software ecosystem very agile,