Large Scale Cluster Computing for Comprehensive Risk Assessment

Nathan Crosby, Harry Millwater
University of Texas at San Antonio

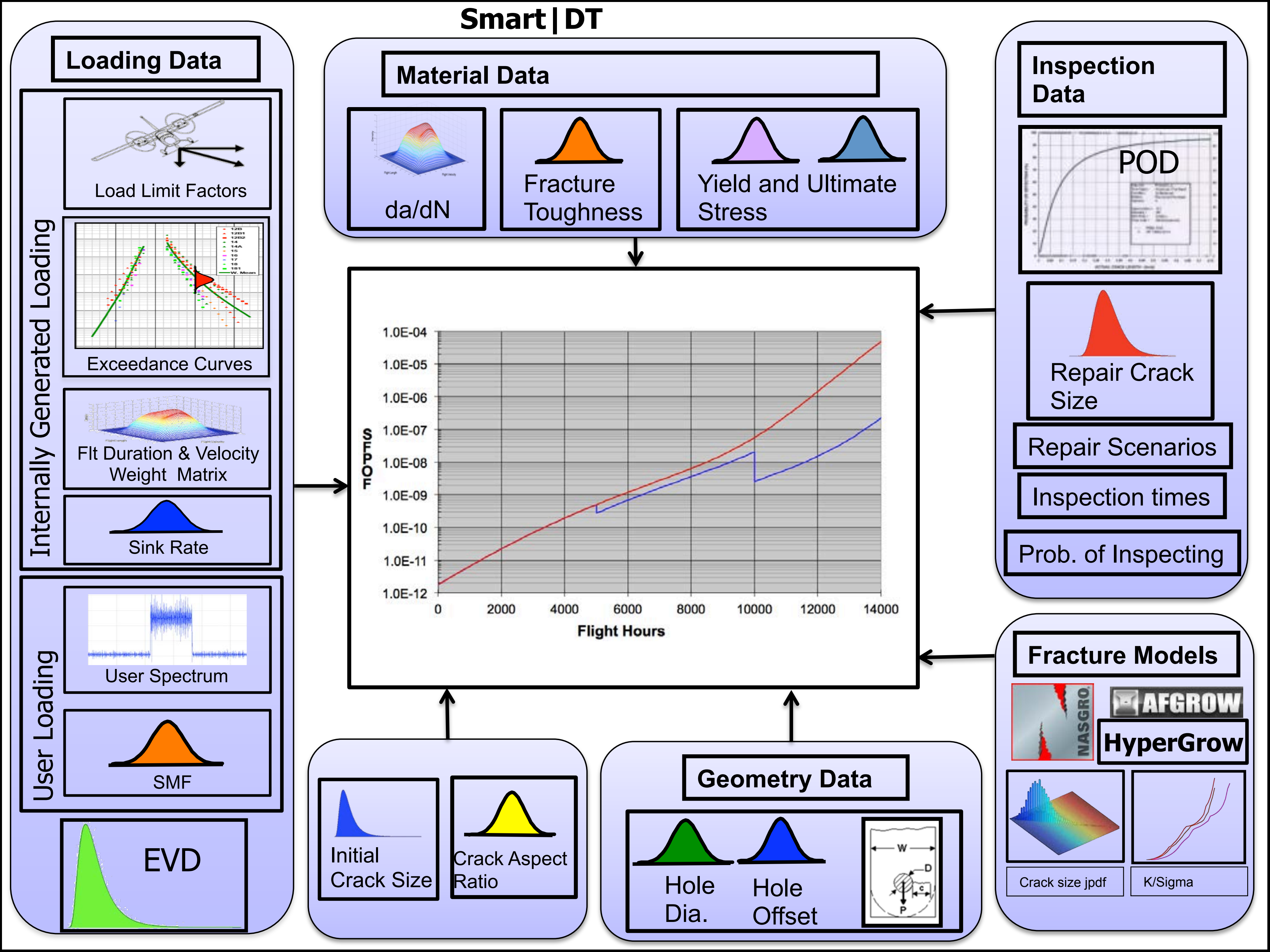
Juan D. Ocampo
St. Mary's University, San Antonio

Beth Glass, Chris Hurst
Textron Aviation

Marv Nuss
Nuss Sustainment

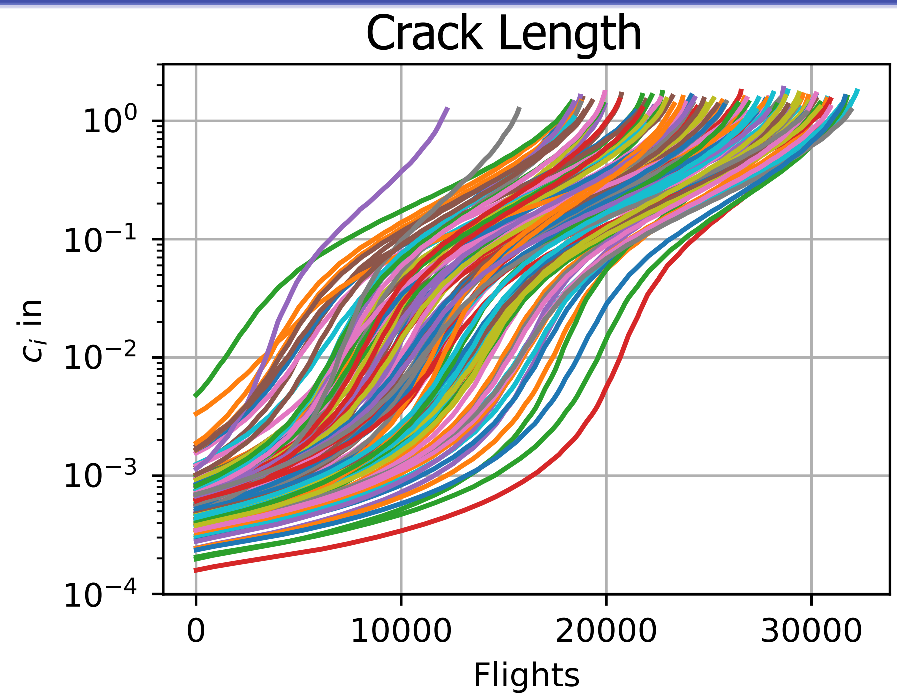
Overview

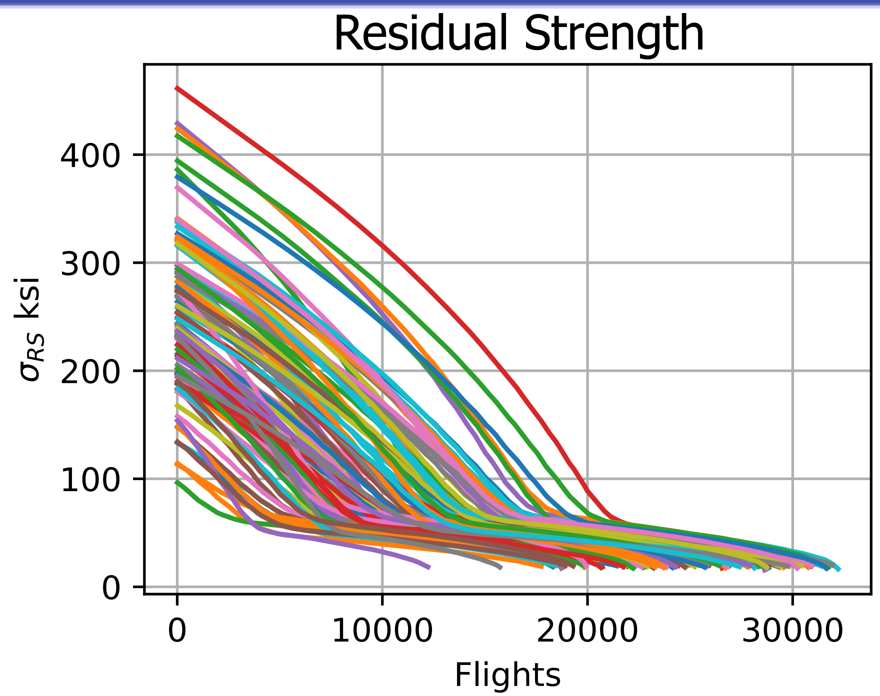
- SMART|DT and Comprehensive Probabilistic Damage Tolerance Analysis (PDTA)
- Large Scale Computing
- Example: Through Crack in Fastener Hole
- Conclusions and Future Work



Smart | DT
Loading Data
Internally Generated Loading
Load Limit Factors
Exceedance Curves
Flt Duration & Velocity
Weight Matrix
Sink Rate
User Loading
User Spectrum
SMF
EVD
Material Data
da/dN
Fracture Toughness
Yield and Ultimate Stress
Inspection Data
POD
Repair Crack Size
Repair Scenarios
Inspection times
Prob. of Inspecting
SFPOF
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08
1.0E-09
1.0E-10
1.0E-11
1.0E-12
0
2000
4000
6000
8000
10000
12000
14000
Flight Hours
Fracture Models
NASGRO
AFGROW
HyperGrow
Crack size jpdf
K/Sigma
Initial Crack Size
Crack Aspect Ratio
Geometry Data
Hole Dia.
Hole Offset
W
D
c
P

Challenge of Comprehensive PDTA

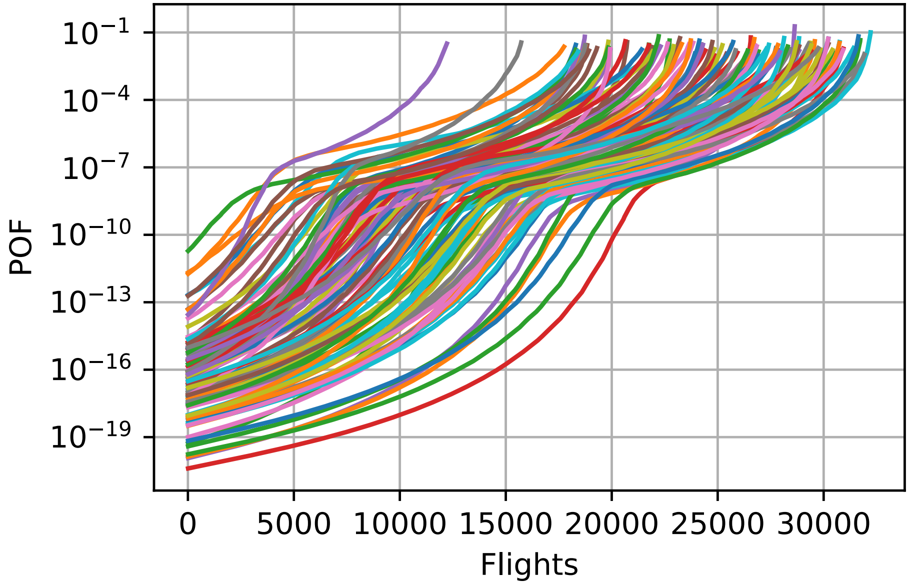


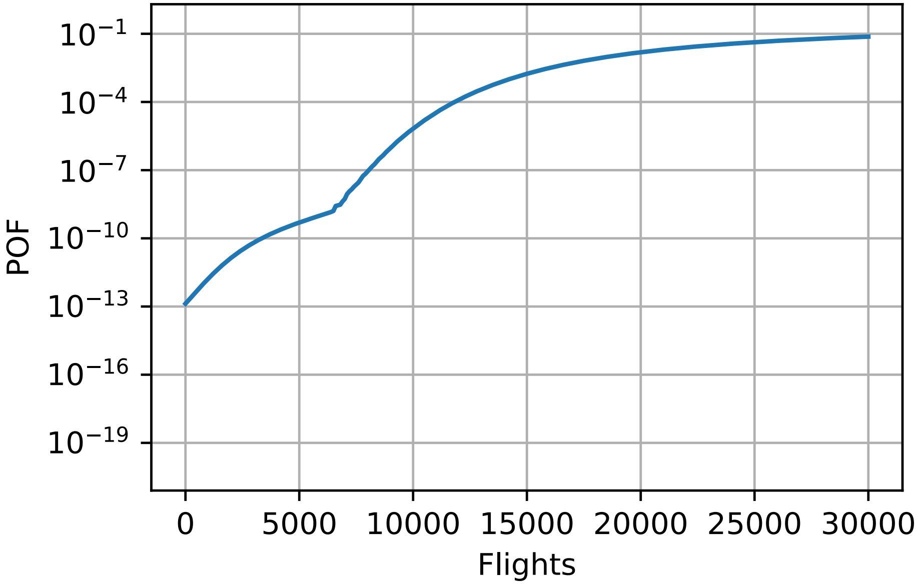


- Comprehensive PDTA requires evaluating the crack growth curve for every realization
- To estimate POF < 10^{-7}, Monte Carlo integration requires minimally 10^8 realizations
- O(10) seconds per realization for 10^8 realizations
  - at least 30 years on a 10 core computer
  - at least 3 year on a 100 core cluster
  - at least 3 months on a 1000 core supercomputer

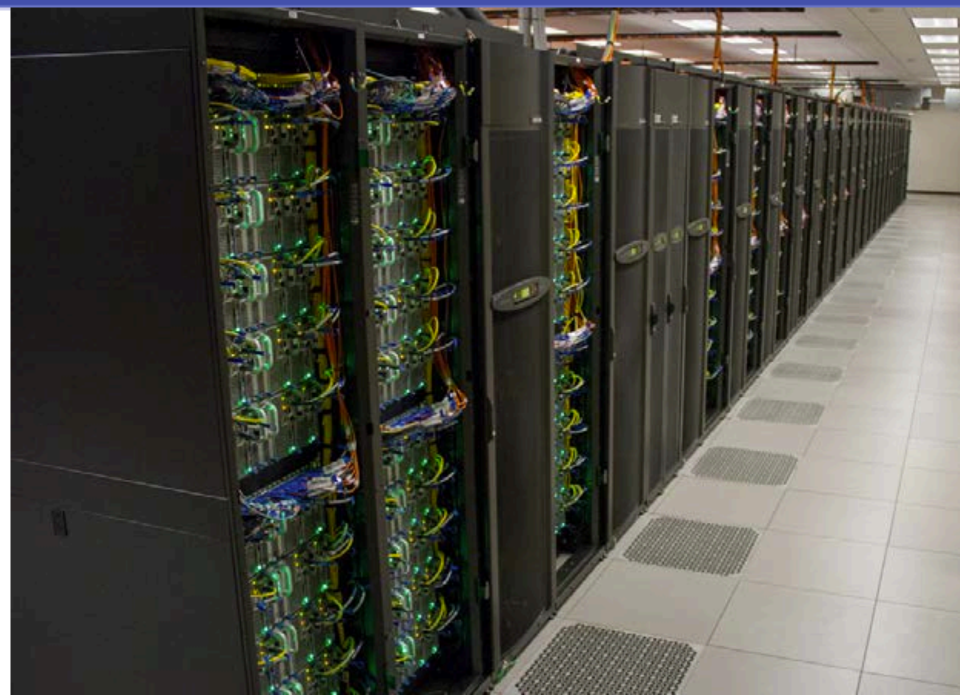
Motivation

- Verification – check results of fast methods against 10^8+ runs of conventional crack growth codes
- Capability – increase the speed of fast methods by 100x or more
- Access – give SMART|DT users the option to leverage large cluster / cloud systems





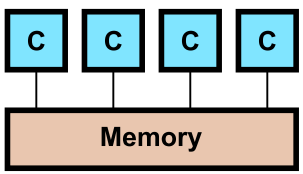
Large Scale Computing



- Large scale computing systems are becoming more available
- High Performance Computing cluster configurations can be configured and used on a pay as you go basis at the 3 major cloud computing providers
- Coding for distributed memory parallelization is required to take advantage of these systems
  - More details in the next slides

Large Scale Computing

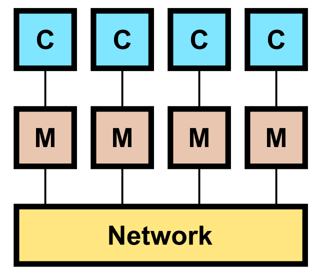
- Shared Memory (OpenMP)
  - communication via shared variables



2 cores

16 cores

- Distributed Memory (MPI)
  - communication via message passing over network

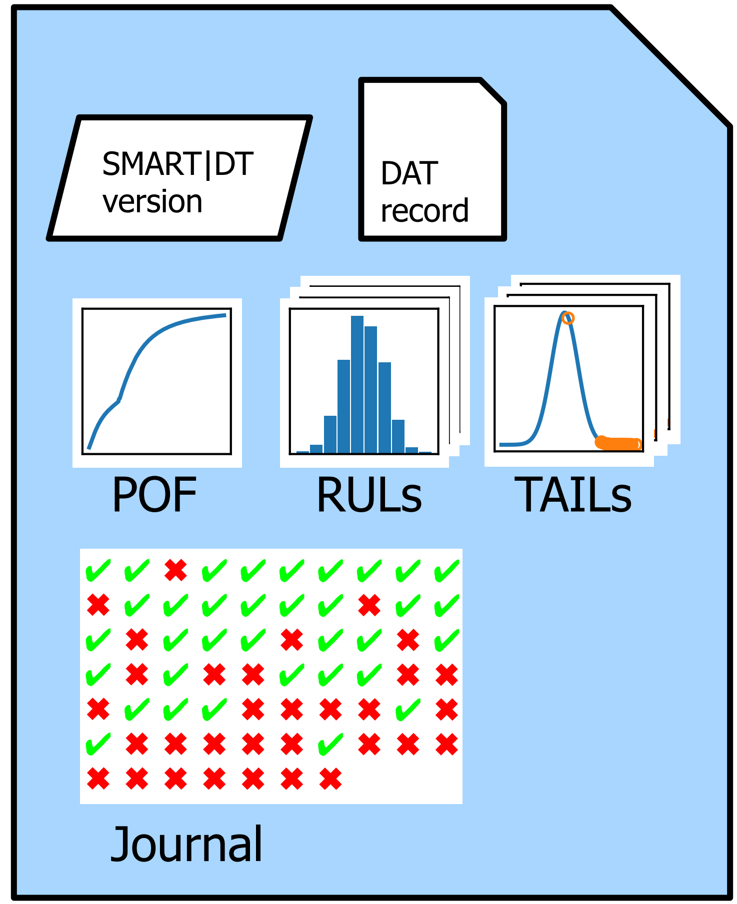


320 cores

102400 cores

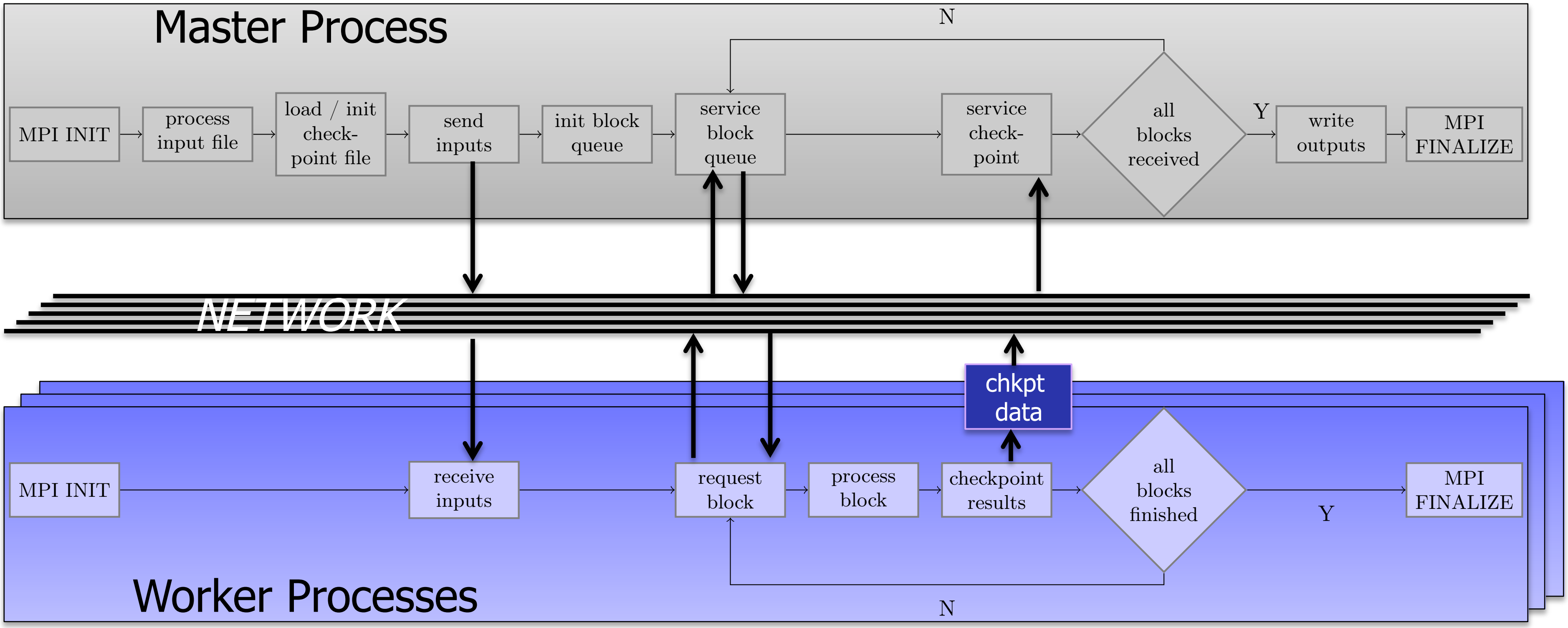
Checkpointing Capability

- Large scale computing incurs an increased risk of system faults
- Checkpoints allow jobs to be restart from somewhere in the middle
- Restarting requires some consistency checks in case files have been modified or the SMART|DT version has changed



jobname.chkpt

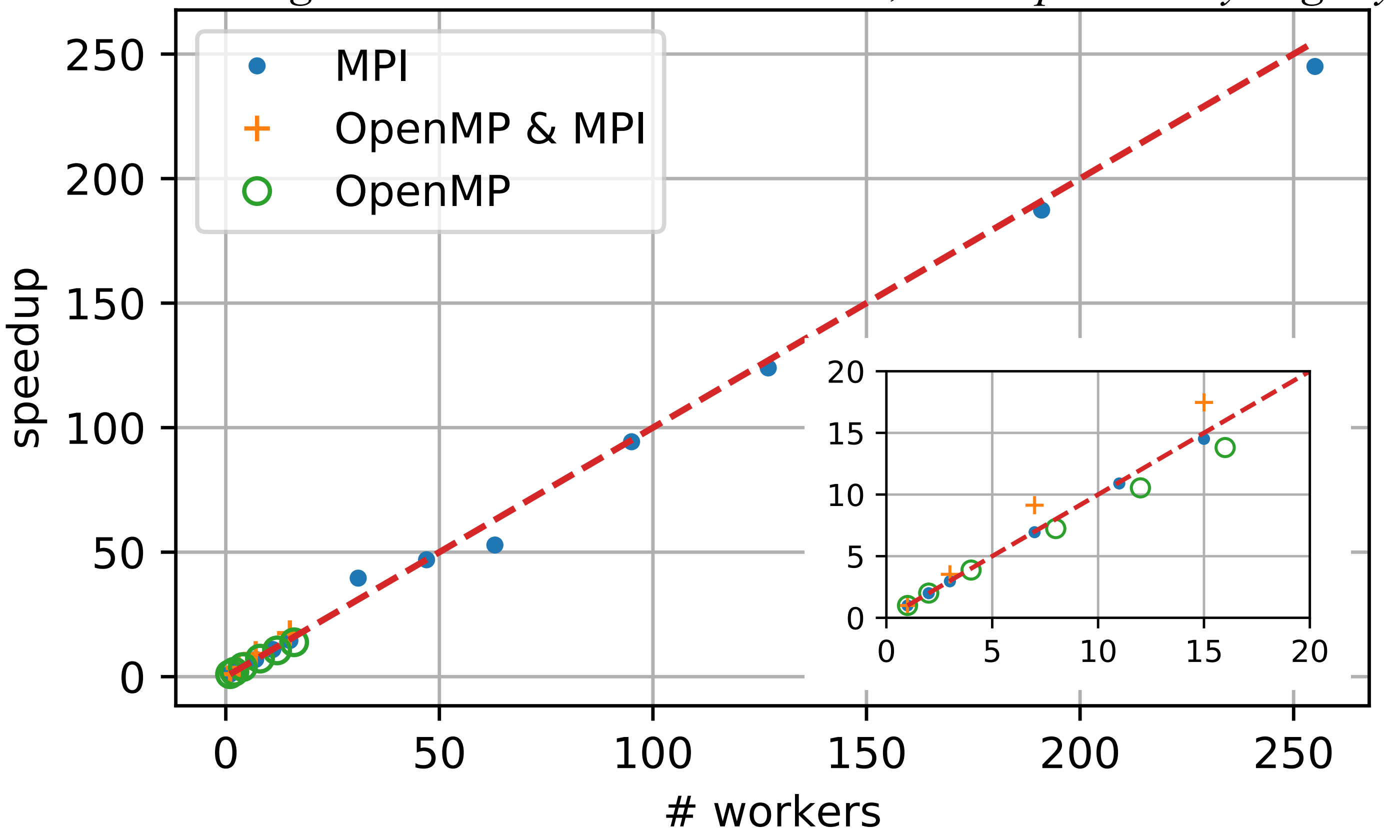
SMART|DT MPI overview



- Master process performs all of the input reading and checking once and sends processed inputs to the workers
- Small data packets exchanged to assign blocks and report block processing complete
- Checkpoint data is buffered so the worker can start processing the next block while the results are transferred

SMART|DT MPI Scaling

Scaling measured on Shamu at UTSA, node speeds vary slightly

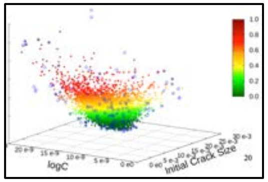


- Near ideal scaling for 250+ processes running HyperGrow realizations

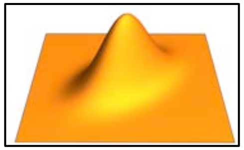
Approaches to Speed Up PDTA

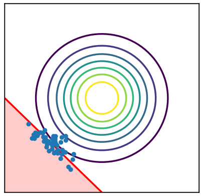
- Run realizations in parallel
  - Large scale computing (uses lots of resources)

- Reduce cost of each realization
  - Master Curve (limited to 3 RVs)
  - HyperGrow (no retardation)
  - Surrogate Model (adds complexity and uncertainty)

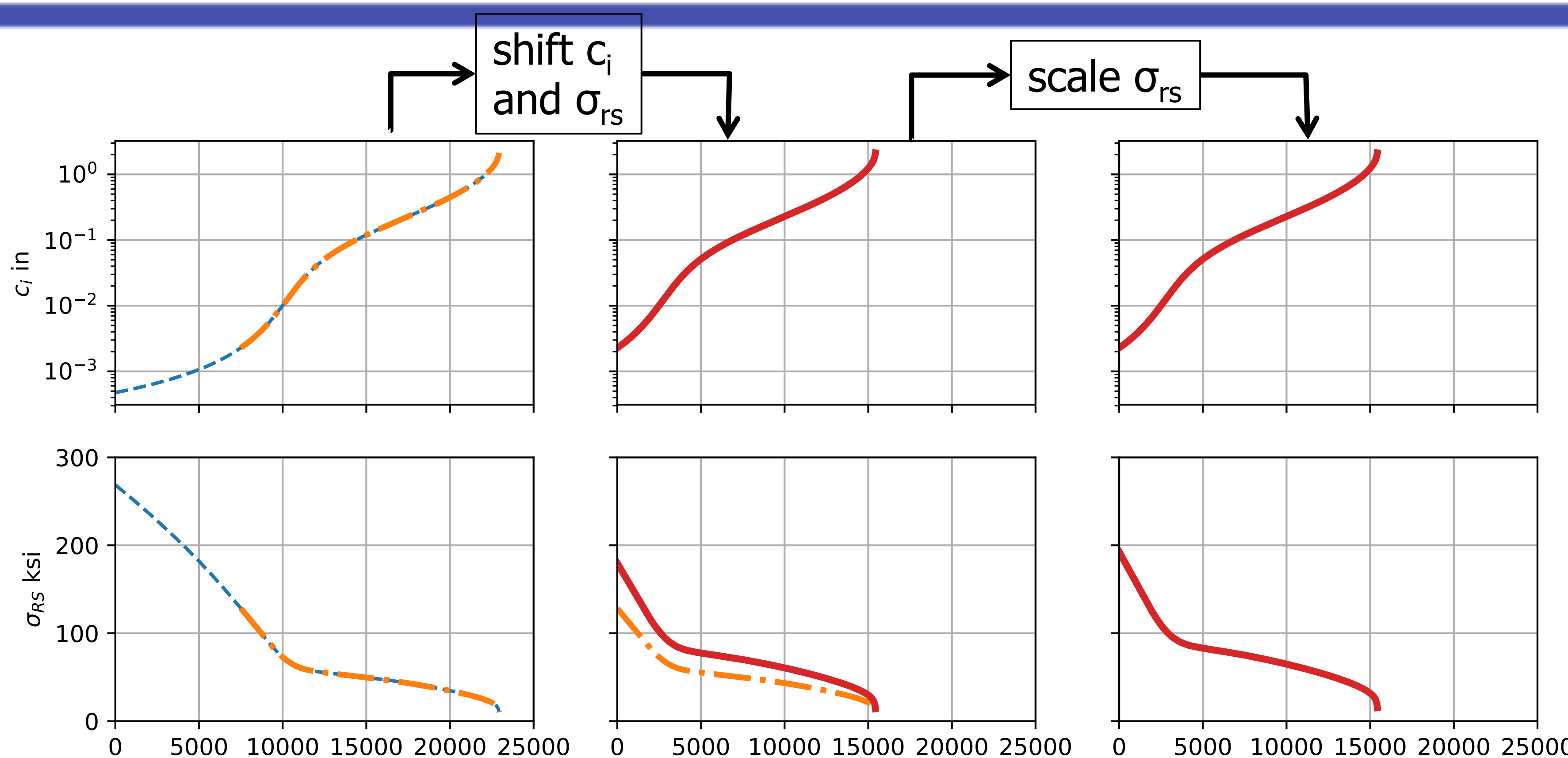


- Reduce total number of realizations
  - Numerical Integration (low dimensions)
  - FORM / SORM (many assumptions)
  - Importance Sampling (requires a priori knowledge)



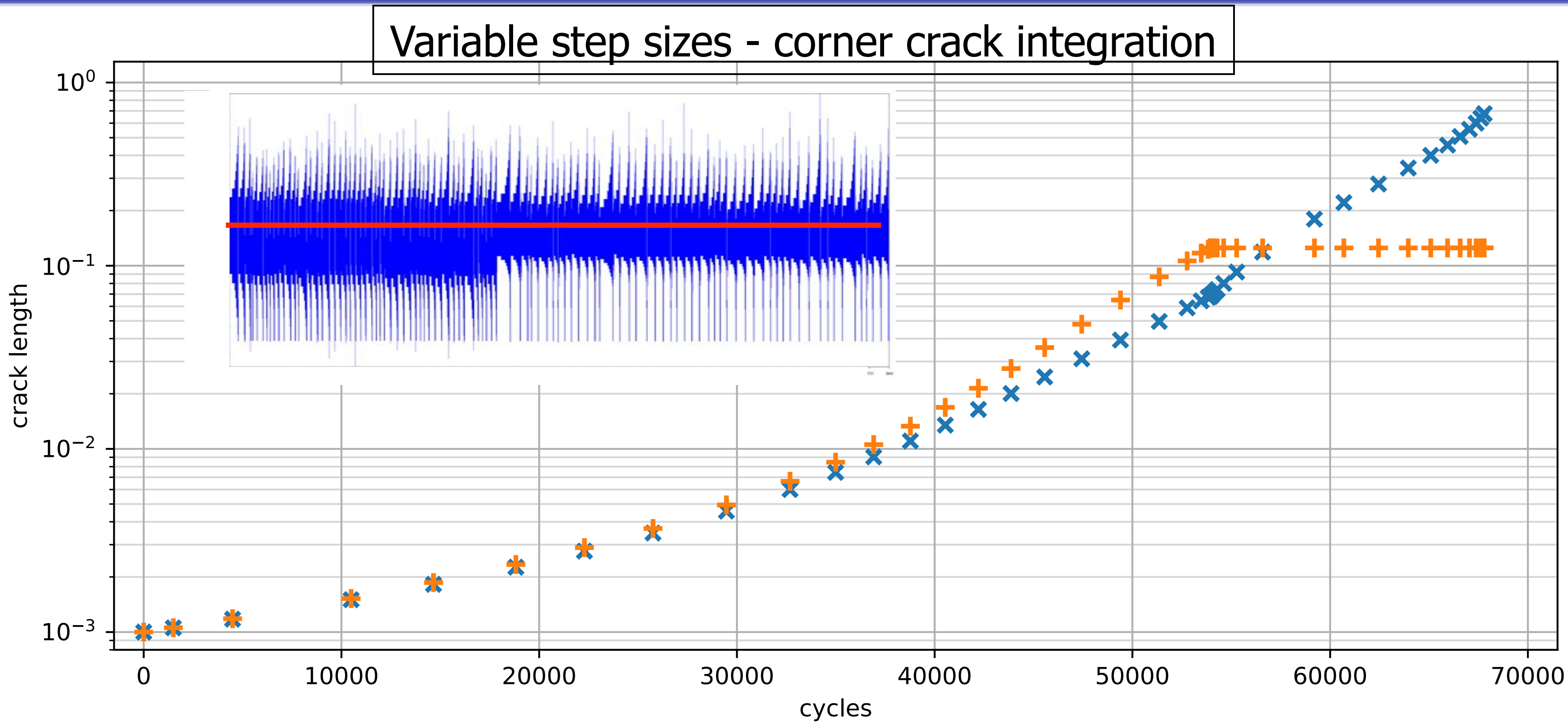


Master Curve



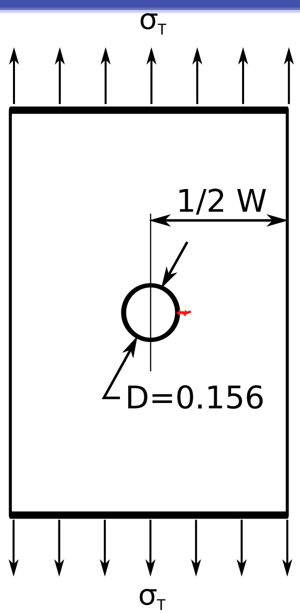
- Ignores variation in crack growth rate, component geometry, etc…

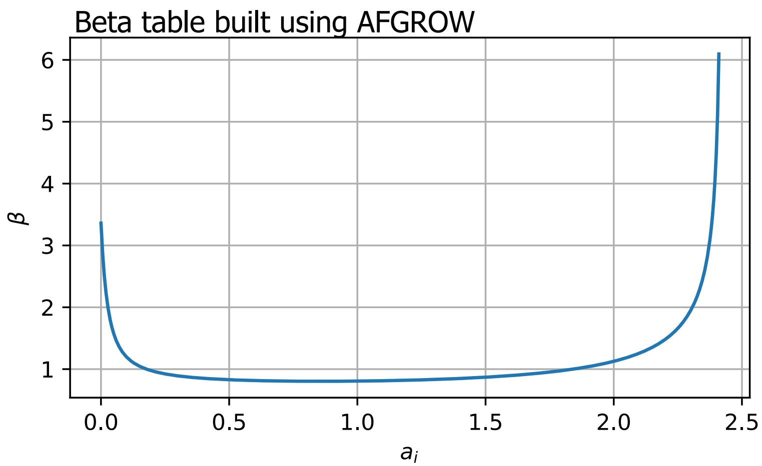
HyperGrow

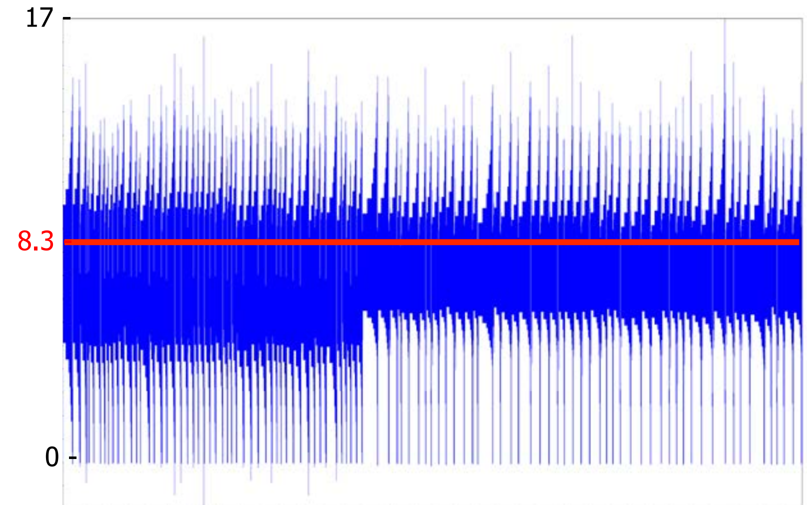


- Calculates Equivalent Constant Amplitude Stress for a given spectrum and uses variable step size integration for significantly faster crack growth evaluation
  - >10000x faster than cycle-by-cycle integration

Example Problem



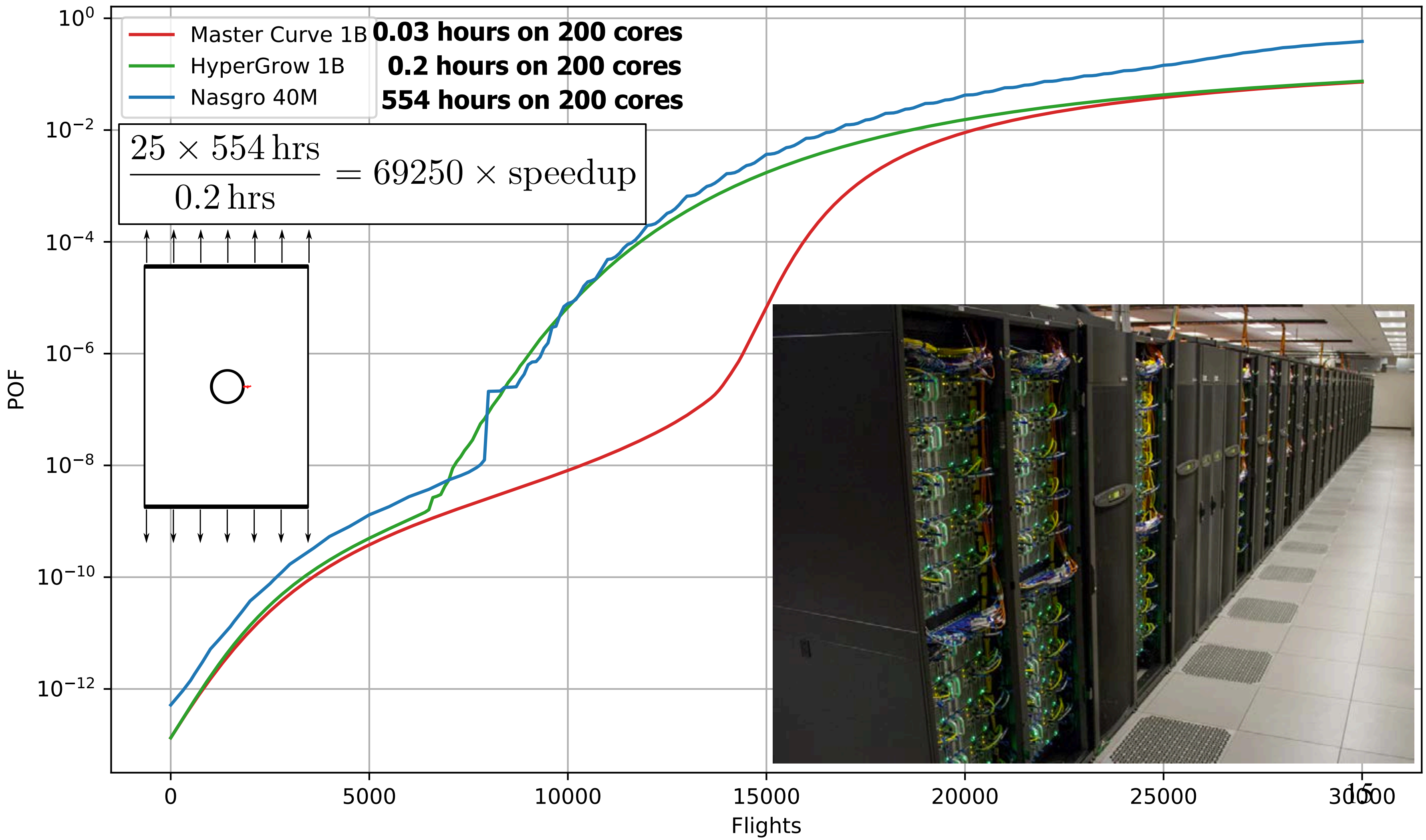




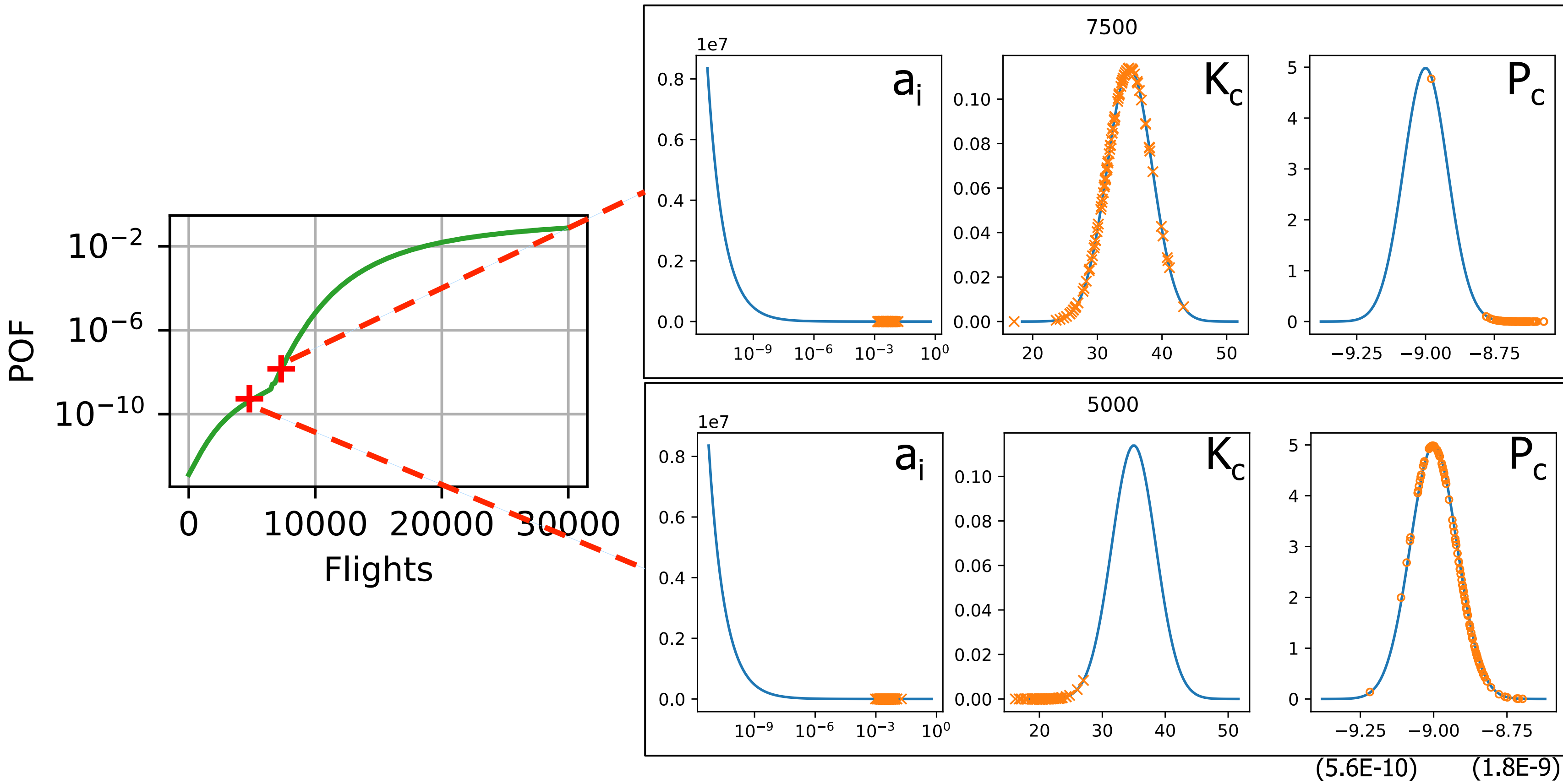
Geometry	Deterministic(5 wide x 0.125 thick)	in
Initial Crack Size	Weibull(α=0.45, β=4.17e-5)	in
Fracture Toughness	Normal(μ=35.0, σ=3.5)	ksi √in
log(Paris C)	Normal(μ=-9.0, σ=0.08)	
Paris n	3.8	
Maximum Load	Frechet(μ=13.4, σ=1.29, ξ=0.07)	ksi

UTSA ROADRUNNERS NuSS Sustainment Solutions
TEXTRON AVIATION

Uninspected POF Output

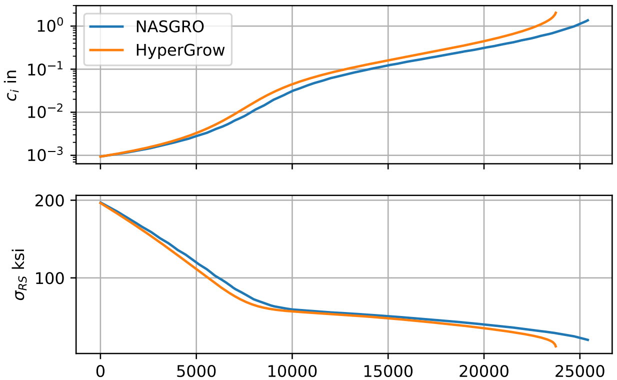


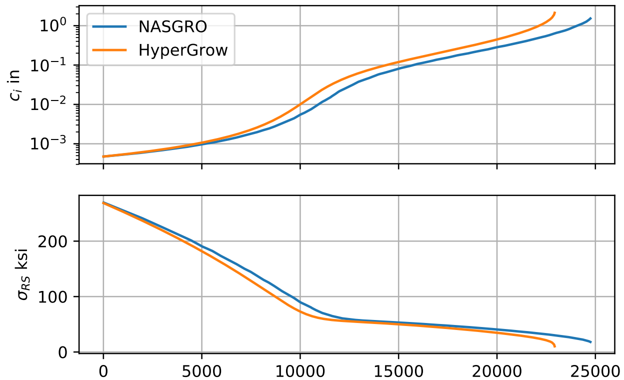
Most Influential Realizations

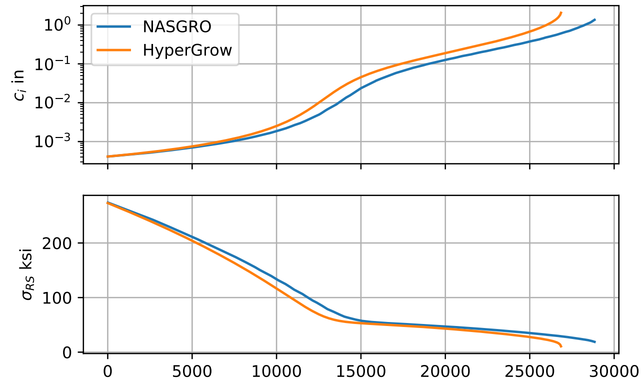


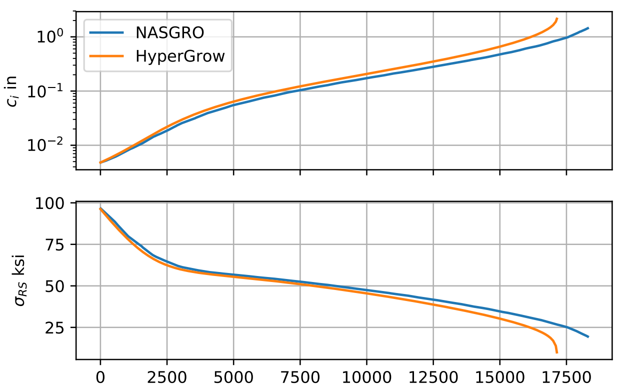
- Visualizes which realizations contribute most to POF

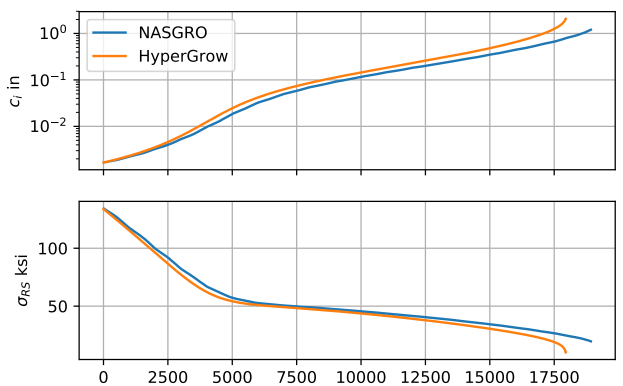
AVSN comparisons - NASGRO and HyperGrow

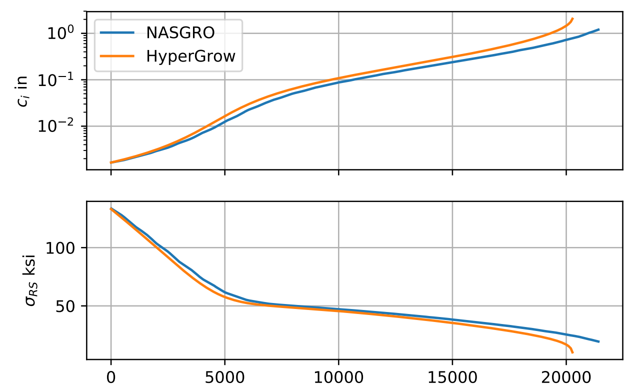




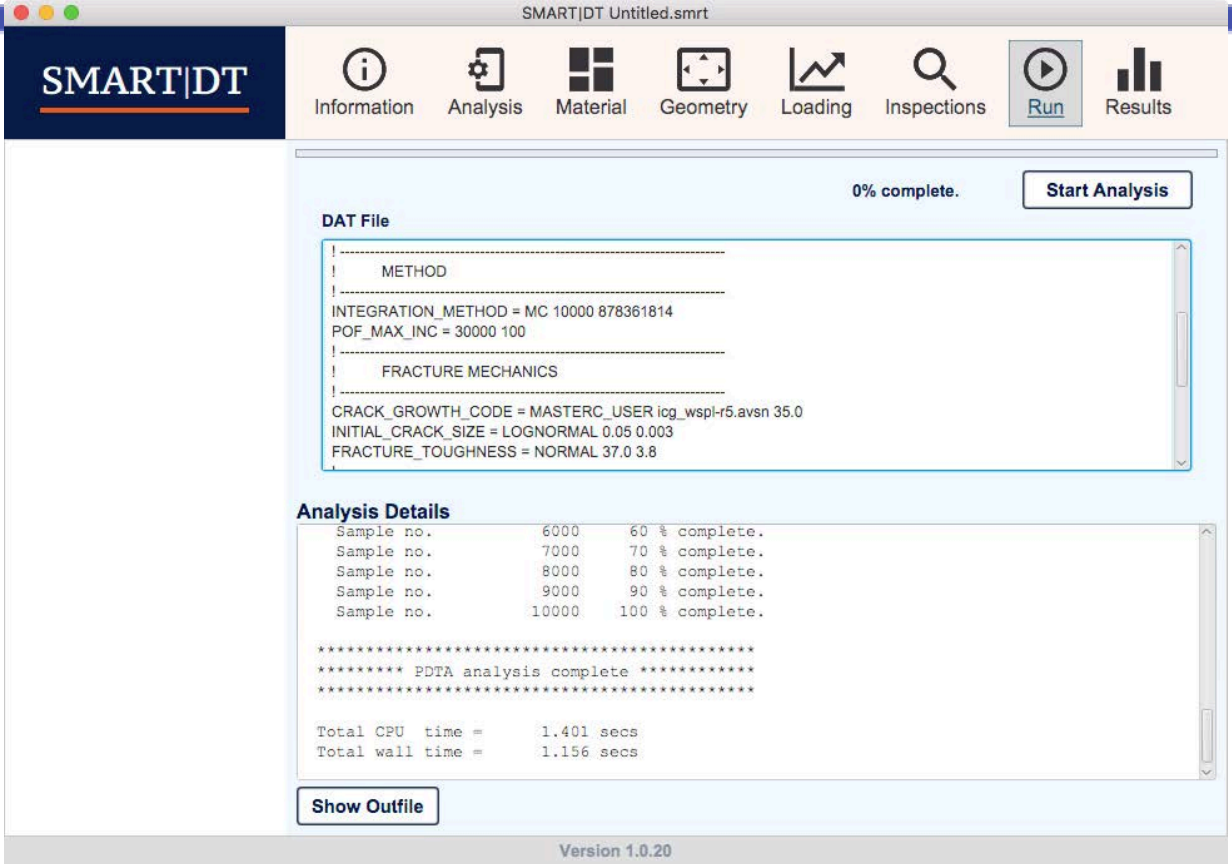








Future Work



- SMART|DT GUI
- Continue HyperGrow Development and Verification
- Add FORM and Importance Sampling Methods

Conclusions

- Large scale computing is necessary to verify methods that speed up the fracture mechanics or probabilistic method

- The combination of HyperGrow and Large Scale Computing makes comprehensive PDTA attainable

Acknowledgments

- Cluster version of SMART|DT funded by UTSA OpenCloud Institute

- This work received computational support from UTSA's HPC cluster Shamu, operated by the Office of Information Technology.

- SMART|DT was developed under FAA grant 16-G-005

Thank you