Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

Software Architecture: Where are we headed?

Jim Kowalkowski

FIFE Workshop

June 20-21, 2016

Remarks

- This talk will
  - Discuss future directions, including a roadmap
  - Highlight a few project you may not know about
  - Discuss a few interesting features you may see
- The laboratory operations review happened in May 2016, many of these slides are from my R&D talk at the computing operations review sessions

Software R&D within SCD

- Incrementally evolve software systems, development methods, and toolkits to productively utilize future computing platforms and facilities.
  - Broaden the community of scientists that FNAL works with, developing new working relationships inside and outside HEP
  - Incorporate software and ideas that have been developed, are emerging, or are widely used in other communities
- Move physics algorithms and software infrastructure towards extreme-scale and commercial computing platforms.
  - Advance computing utilizing more specialized computing machinery.
  - Addressing the expected end of Moore's law, predicted to be 2020-2021.
  - Greatly expands the available compute cycles
- Core team includes scientists, computer science researchers, and software developers

New initiative: Code modernization

- Moves us toward future specialized platforms
  - Big data technology where calculations are moved to the data
  - Many-core compute platforms
    - 1000s of cores, large vector units, deep memory hierarchies, high-performance networking, limited disk bandwidth
  - New methods of developing algorithm code
- Touches software toolkits and systems, in areas such as
  - Software frameworks
    - I/O, scheduling, configuration, data access, metadata handling
  - Physics analysis toolkits
    - interactive analysis and visualization
  - Algorithms that operate within the frameworks
  - Workflow integration tools
    - overall campaign management and coordination

R&D Roadmap

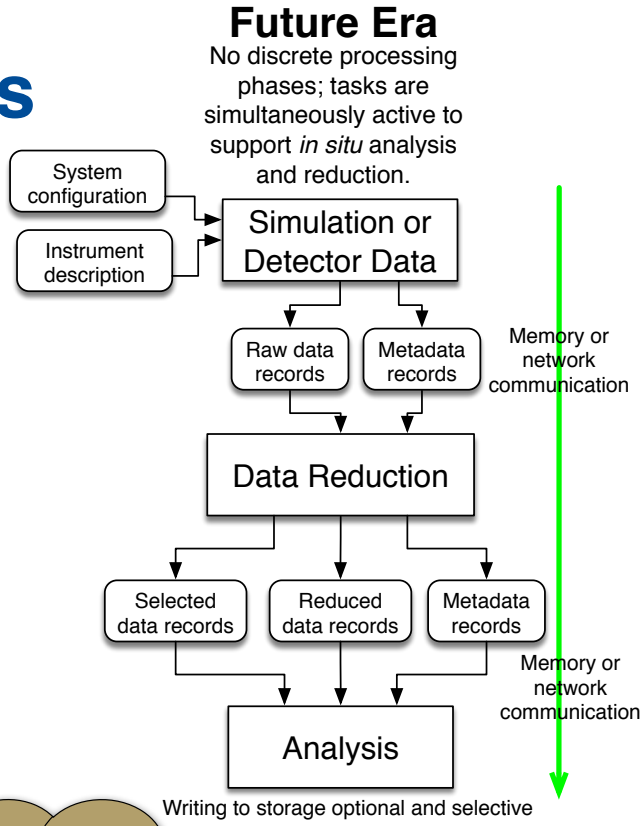
R&D Roadmap	2016		2017		2018		2019-2020		2021-2024		2025-
frameworks - MT	basic event-level parallelism		consistency with CMSSW		GeantV integration						
frameworks - HPC	Integrate Geant4-MT	art HPC upgrades	Mira scaling tests		New architecture - Exascsale distributed art workflows R&D				Integration and preparations for experiments		
frameworks - I/O	HDF5 exploration		Distributed art workflow R&D								
advanced tech - big data	Pilot - CMS Dark Matter Spark use case		studies of LHC Run 3 analysis scaling								
post moore	Automata processor - CMS upgrade tracking				quantum computing						
machine learning			studies for application in HL-LHC and LArTPC								
GeantV	vectorization and MT with CERN				production version						
Visualization	Paraview pilot	framework API upgrade	Visualization toolkit for LArTPC & muon program, working with HPC Paraview projects								
build & release	SpackDev - packaging modernization		integration with HSF								
containers	Cori - MicroBooNE MC	HPC HEP cloud pilot	HPC Release & I/O management								
performance	LArSoft algorithms - CPU & memory			Parallel LArSoft algorithms							
Neutrino experiments	MicroBooNE				ProtoDUNE / ICARUS		SBND				DUNE
LHC							Run 3				HL-LHC
Facilities	Cori Phase I	Cori Phase II	Theta		Aurora				Exascale		

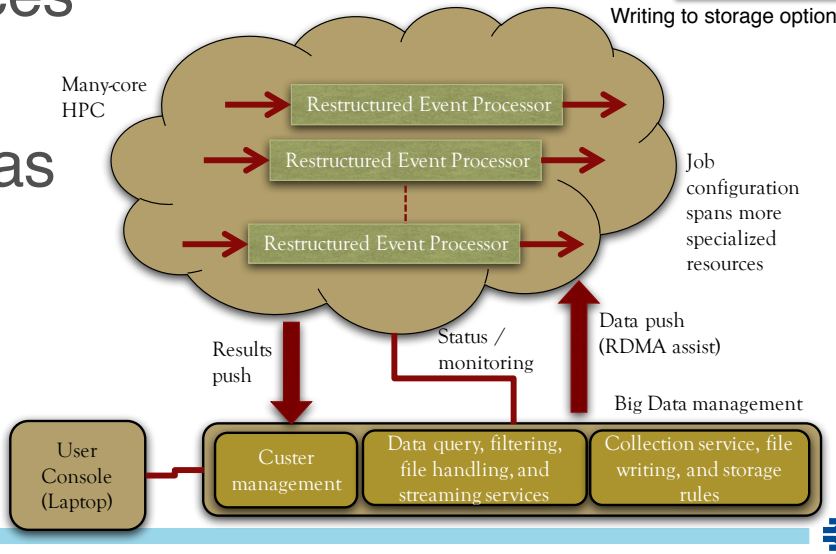
Forward looking | Meeting current needs | Transformative | Experiments & Facilities

- Addresses features that users need
- Includes partnerships with ASCR, national labs, CERN
- Matches experiment and facilities schedules
- Includes evolutionary and transformational changes

Evolving architecture key elements

- Limit I/O to disk storage
- Utilize large vector units
- Efficient movement of data between processes in a distributed environment using high bandwidth networking
- Shared framework services within nodes
- Localized data caching, as in big data technology
- Tighter integration with workload / workflow management



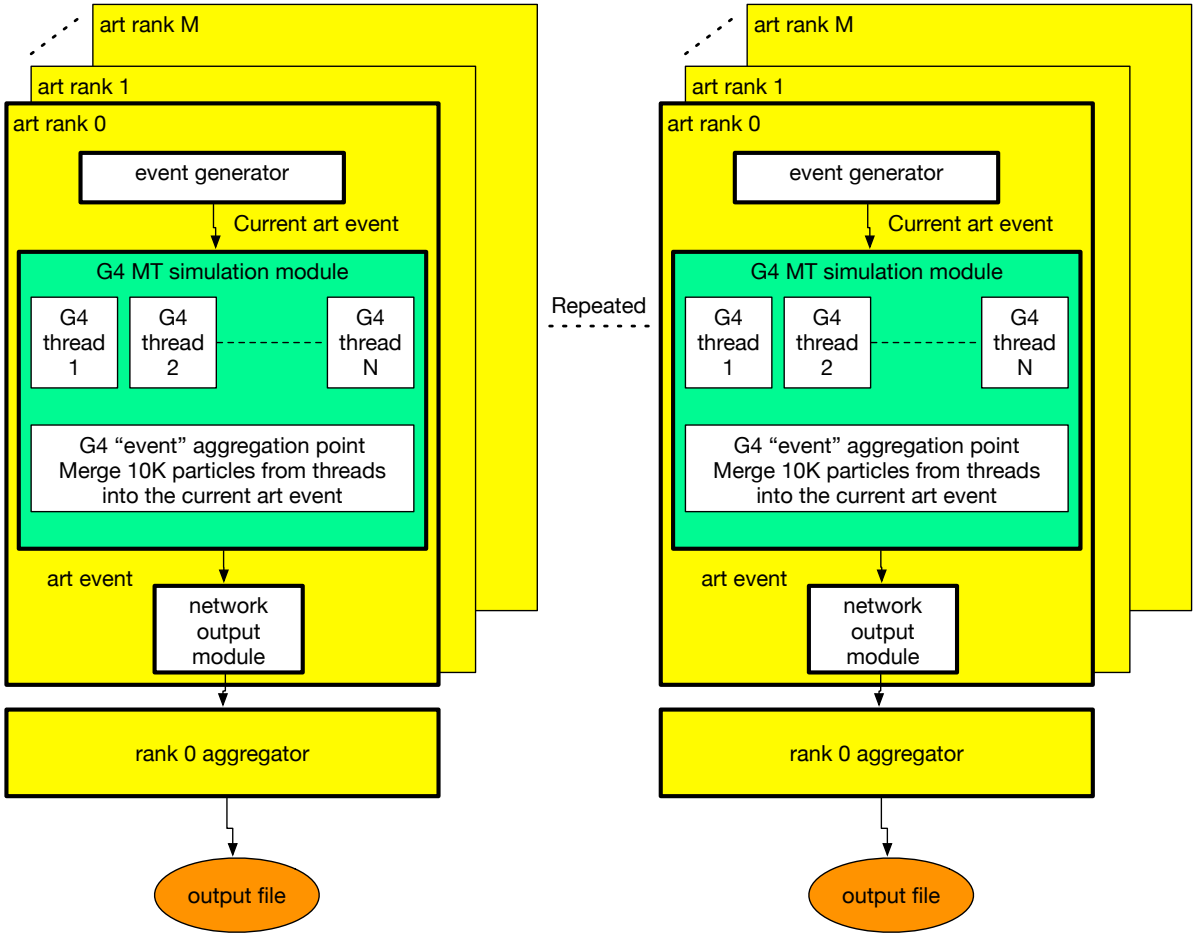


Key forward-looking projects

- Art advanced I/O with HDF5
- Art multi-threading
  - Enabling data structure sharing and parallelism
  - Working together with CMSSW developers http://indico.cern.ch/event/304944/contributions/1672502/
- Containers using Docker and Shifter on NERSC Cori
  - **SCD has enabled Shifter technology in the Intensity Frontier for MicroBooNE** and has provided this as an HPC MiniApp through the FCE, co-lead by FNAL and ANL
  - https://cd-docdb.fnal.gov:440/cgi-bin/ShowDocument?docid=5693
- Advanced visualization with ParaView
  - https://cd-docdb.fnal.gov:440/cgi-bin/ShowDocument?docid=5647
- Big Data
  - CMS analysis using Spark
  - Using HDF5 and Spark on Cori
- Includes Art-HPC, GeantV, and pythia-on-Mira projects

art-HPC

- **Extending the ART Framework to Support Large Scale Multiprocessing for the Intensity Frontier**
  - Partnership with Tom LeCompte at ANL
  - Migration of art to HPC and Mira
  - Using MPI
  - Multi-threaded Geant4
- Target is to produce 10^12 muons for muon g-2 on ALCF Mira
- Architected to addressed
  - limit I/O to filesystem
  - scaling

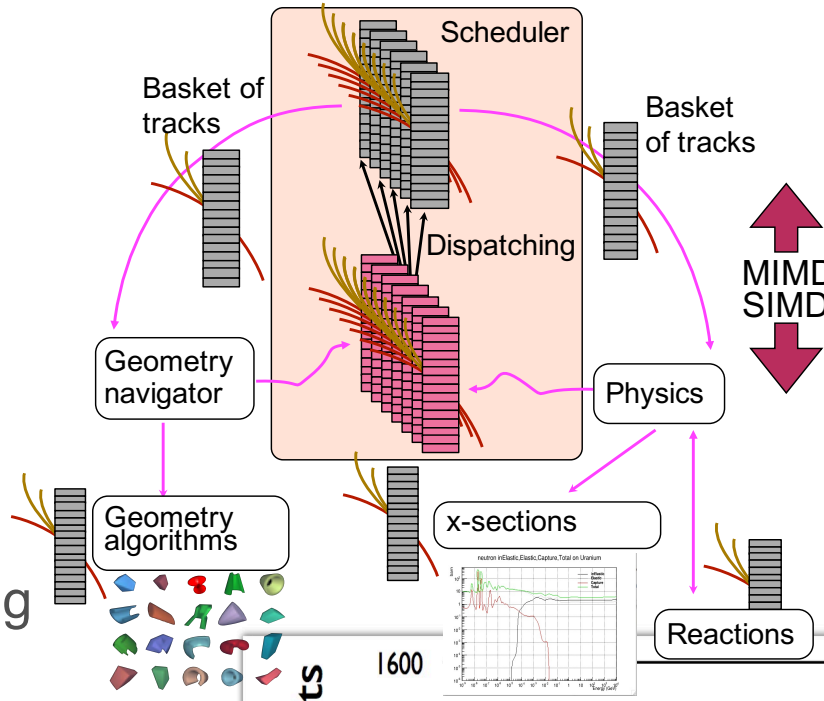


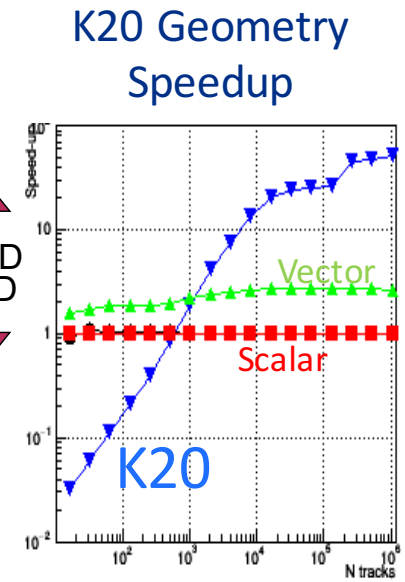
NOTE: Same architecture applied to running a multi-parameter tuning of event generators using collider data analysis on Mira using Pythia

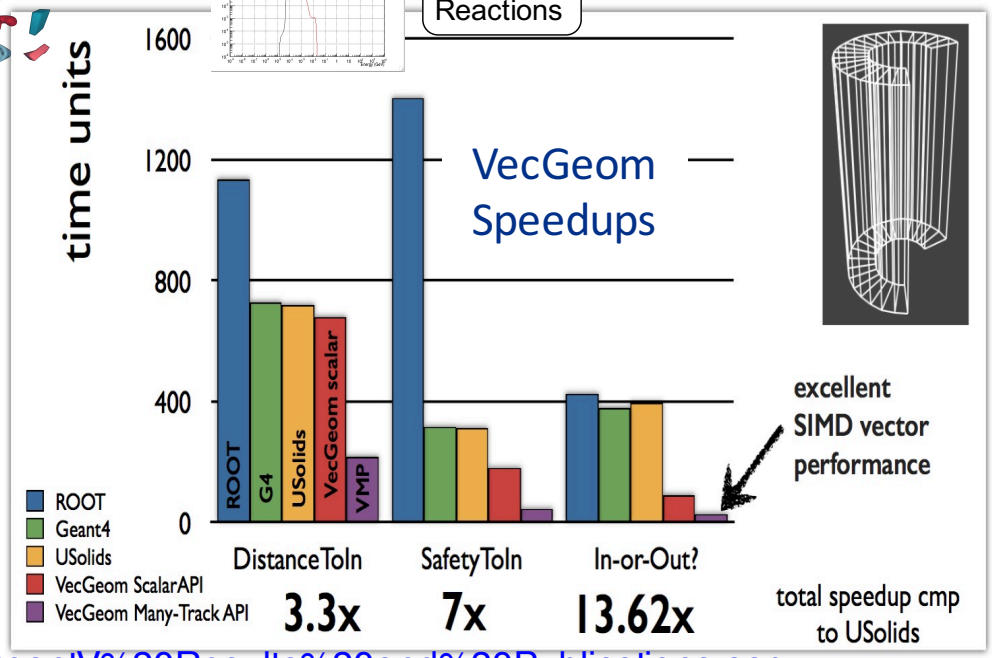
https://cdcvs.fnal.gov/redmine/projects/art-hpc/wiki/

GeantV

- R&D Partnership with CERN
- ASCR aided in performance analysis early on
- Developed package **VecGeom** already being utilized
- Incorporating MIC, GPU, and multi-core
- Plans for integration with CMSSW and later with art
- Architecture targets use of vector units







https://web.fnal.gov/project/Geant4RD/SitePages/GeantV%20Results%20and%20Publications.aspx

Summary

- Mainstream architecture is evolving to match the facility changes and future experimental needs
  - Pulling from tight connections with CMSSW
  - Utilizing broad community tools
    - Docker, Paraview, MPI, numpy/scipy, HDF5, Spark, Spack
  - Collaborating on Big Data and Machine Learning forums.
- A code modernization initiative is in the future
  - Increase resource utilization using large vector units
  - Shared data structures through multi-core computing
  - Distributed memory and workflow
- Collaborative efforts are expanding
  - ASCR, HEP CCE, FCE, HSF
  - ANL leadership facility, NERSC
  - CERN openlab and software groups
- We will remain consistent with laboratory strategic themes and addressing risks.

Backup slides

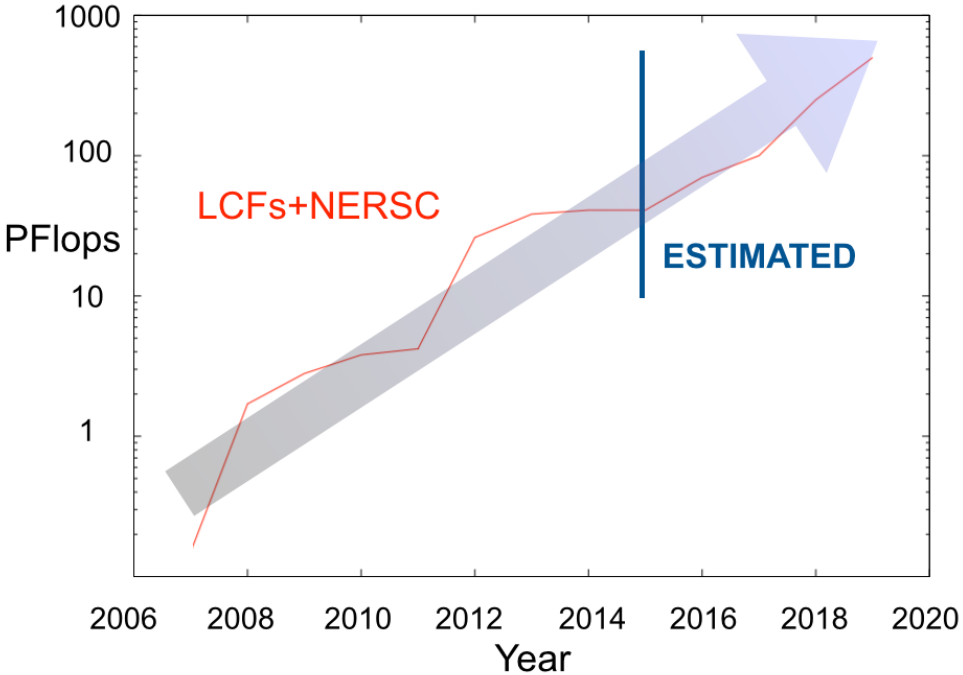
Some relevant factors heading into the future

- MicroBooNE wants to be MT by 2020 (or so).
- DUNE wants to look into a whole new framework starting in 2018. We want to be ready to meet their list of requirements by then.
- HPC resources within HEPcloud will almost certainly be whole-node scheduled.
- AWS resources within HEPcloud will almost certainly be most easily handled using whole-node scheduling.
- Our use of HPC resources is likely going to be tied to containerization.

From FCE meeting: The Impact of HPC on HEP

- Not all problems can be solved using HPC systems, but many can (accelerators, cosmology, event generation/simulation, QCD…)
- Next generation of ASCR HPC machines (staging begins 2016, ends in 2018) will sum to ~500 petaflops of compute capability

- If HEP experiments use just 5% of that, i.e. 25 petaflops, it is ~25 times what the Grid will provide
- Learning how to leverage these resources to seamlessly supplement/enhance current capability is important
- New possibilities opened up by HPC platforms will offer unique computational opportunities



ASCR resources

ASCR Computing At a Glance

now ← | → future

System attributes	NERSC Now	OLCF Now	ALCF Now	NERSC Upgrade	OLCF Upgrade	ALCF Upgrades	
Name Planned Installation	**Edison**	**TITAN**	**MIRA**	**Cori 2016**	**Summit 2017-2018**	**Theta**	**Aurora 2018-2019**
System peak (PF)	2.6	27	10	> 30	150	>8.5	180
Peak Power (MW)	2	9	4.8	< 3.7	10	1.7	13
Total system memory	357 TB	710TB	768TB	~1 PB DDR4 + High Bandwidth Memory (HBM) +1.5PB persistent memory	> 1.74 PB DDR4 + HBM + 2.8 PB persistent memory	>480 TB DDR4 + High Bandwidth Memory (HBM)	> 7 PB High Bandwidth On-Package Memory Local Memory and Persistent Memory
Node performance (TF)	0.460	1.452	0.204	> 3	> 40	> 3	> 17 times Mira
Node processors	Intel Ivy Bridge	AMD Opteron Nvidia Kepler	64-bit PowerPC A2	Intel Knights Landing many core CPUs Intel Haswell CPU in data partition	Multiple IBM Power9 CPUs & multiple Nvidia Voltas GPUS	2nd gen Intel Xeon Phi processor (code name Knights Landing)	3rd gen Intel Xeon Phi processor (code name Knights Hill)
System size (nodes)	5,600 nodes	18,688 nodes	49,152	9,300 nodes 1,900 nodes in data partition	~3,500 nodes	>2,500 nodes	>50,000 nodes
System Interconnect	Aries	Gemini	5D Torus	Aries	Dual Rail EDR-IB	Aries	2nd Generation Intel Omni-Path Architecture
File System	7.6 PB 168 GB/s, Lustre®	32 PB 1 TB/s, Lustre®	26 PB 300 GB/s GPFS™	28 PB 744 GB/s Lustre®	120 PB 1 TB/s GPFS™	10PB, 210 GB/s Lustre initial	150 PB 1 TB/s Lustre®

U.S. DEPARTMENT OF ENERGY | Office of Science

LANS/LLNS May 29, 20

Advantages of multi-core: Better scaling of system resources as core counts increases

- Application
  - Reduce memory needed per CPU core
  - share infrequently updated memory [conditions, I/O buffers]
  - share file handles
  - share network connections (database and otherwise)
  - No need to read executable code more than once
- Better scaling for sites
  - reduce number of requests to database
  - reduce number of open files
  - reduce number of necessary batch slots (not 1 per core but 1 per node)
  - reduce memory needed per core
- Workflow system
  - reduce time it takes to process 'one atomic unit of processing'
    - [CMS must process all of a LuminosityBlock in just 1 process] this reduces the time for retrying failed jobs which decreases the overall time to finish a workflow
  - reduce number of jobs to be tracked in a workflow
- Running on your own machine is faster!
  - Look at test suite running on woof!

Some limitations on multi-core

- Generally decreases the average throughput for a workflow compared to N single-threaded jobs
- Sites have to deal with a mixture of single and multi-core jobs
- Getting code to become and then stay multi-thread safe is hard with very few tools to help.
- Amdahl's law means quickly diminishing returns with even just a little serialization
  - ROOT might be one of these serialization points!
  - Say we're trying to use a 64 core machine, using one 64-thread job.
    - If the program is 1% serialized, then 64 threads will run 100/(1+99/64) times faster, for a factor of 39.2638 (64 for perfect scaling).
    - If the program is 5% serialized, then the result is 100/(5+95/64) = 15.42169 times faster (rather than 64 for perfect scaling).

Maintaining necessary strengths & skills

- Modern programming with C++14
  - including template meta-programming
  - Task-based programming
- Performance improvements
  - CPU and memory
  - Reviews and recommendations
  - Corrections to algorithms and data structures

Consistency with Fermilab Strategy

- In accord with the Computational Science Theme objectives
  - GL-00510: Modernize scientific processes and access to computing
  - GL-00500: Continuously improve physics and infrastructure
  - GL-00530: Improve the partnership with the DOE Advanced Scientific Computing Research (ASCR) division and help drive the National Strategic Computing
  - GL-00520: Perform R&D to facilitate adoption of industry standards and emerging technologies
- Addressing the highest risk items
  - RT-OPS-CD-001: Inability to leverage computing industry hardware and software advances
  - RT-OPS-CD-002 Insufficient dedicated on-site computing resources
  - RT-OPS-CD-007 Inability to scale services to required capacity
  - RT-OPS-CD-009 Inability to satisfy short-term peak requests for computing resources