AVEC Level-1 Evaluations

Table 1. Level 1 Testing Evaluation Criteria

Level 1 Eval #	Evaluation Criteria
1	Bit reproducibility for restart under identical conditions
2	Solution realism for dry adiabatic flows and simple moist convection
3	High computational performance (8.5 min/day) and scalability to NWS operational CPU processor counts needed to run 13 km and higher resolutions expected by 2020.
4	Extensible, well-documented software that is performance portable.
5	Execution and stability at high horizontal resolution (3 km or less) with realistic physics and orography
6	Lack of excessive grid imprinting

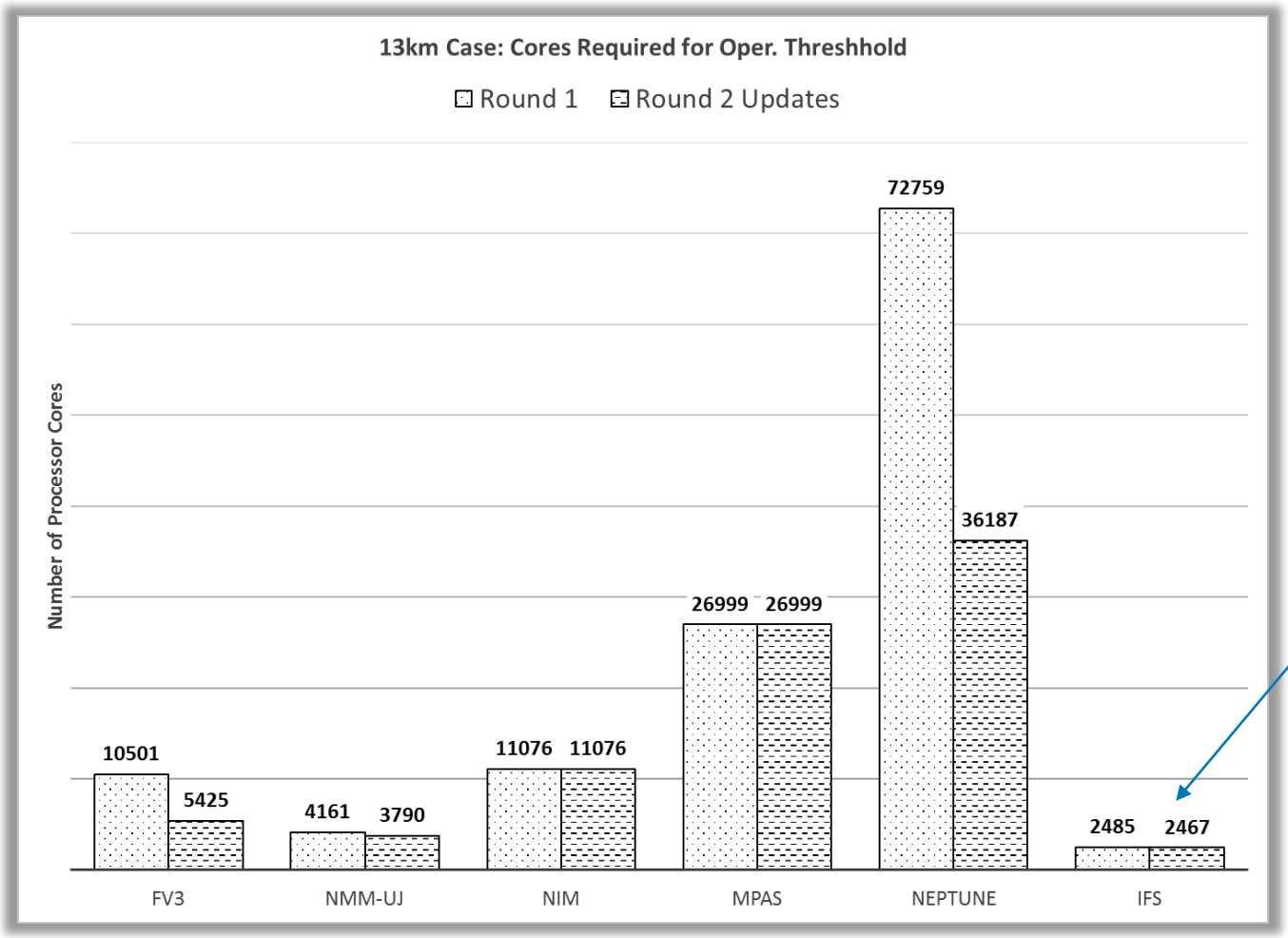
- Performance and Scaling Benchmarks on 130-thousand core DOE supercomputer: "Edison"
  - 13-km and 3-km workloads based on HIWPP non-hydrostatic test case
  - Model groups agreed on each others' configurations
  - Time step and other configuration options were "best guesses"
  - Groups that changed codes or configurations to improve benchmark performance were required to resubmit results for HIWWP test case
- Reports
  - NGGPS Level-1 Benchmarks – April 30, 2015
  - NGGPS Level-1 Software Evaluation (addendum to above) – May 28, 2015

Caveats

- The performance and scaling results in this report are a snapshot in time of NWP software that is under active development. The test workloads are based on an idealized atmospheric case that does not include physics.
- Dedicated access to a 130-thousand core supercomputer is a precious commodity that required emphasizing coverage rather than replication of runs. More replication would better address variability observed in the results from run-to-run and from time step to time step. However, we believe sample sizes were adequate and have discarded obvious outliers in the results.
- The choice of time step for the idealized benchmark runs was best-guess of what would be needed for full-physics real-data forecasts on the part of the modeling groups. In adjusting benchmarking results to the operational speed requirement, we also assumed that dynamics represents half the run time of a full-physics model.
- AVEC did not evaluate important aspects of performance such as I/O, initialization costs, or other factors that would not represent full physics realizations of the models. Such testing will occur in future Level-2 evaluations under the NGGPS test plan.
- AVEC did not evaluate model performance with respect to any objective or subjective measures of solution quality. Each candidate model's benchmarks were conducted with the same formulation and configuration used to run the idealized test cases under the High Impact Weather Prediction Program (HIWPP) non-hydrostatic dycore evaluations.

AVEC Level-1 Evaluations: Performance

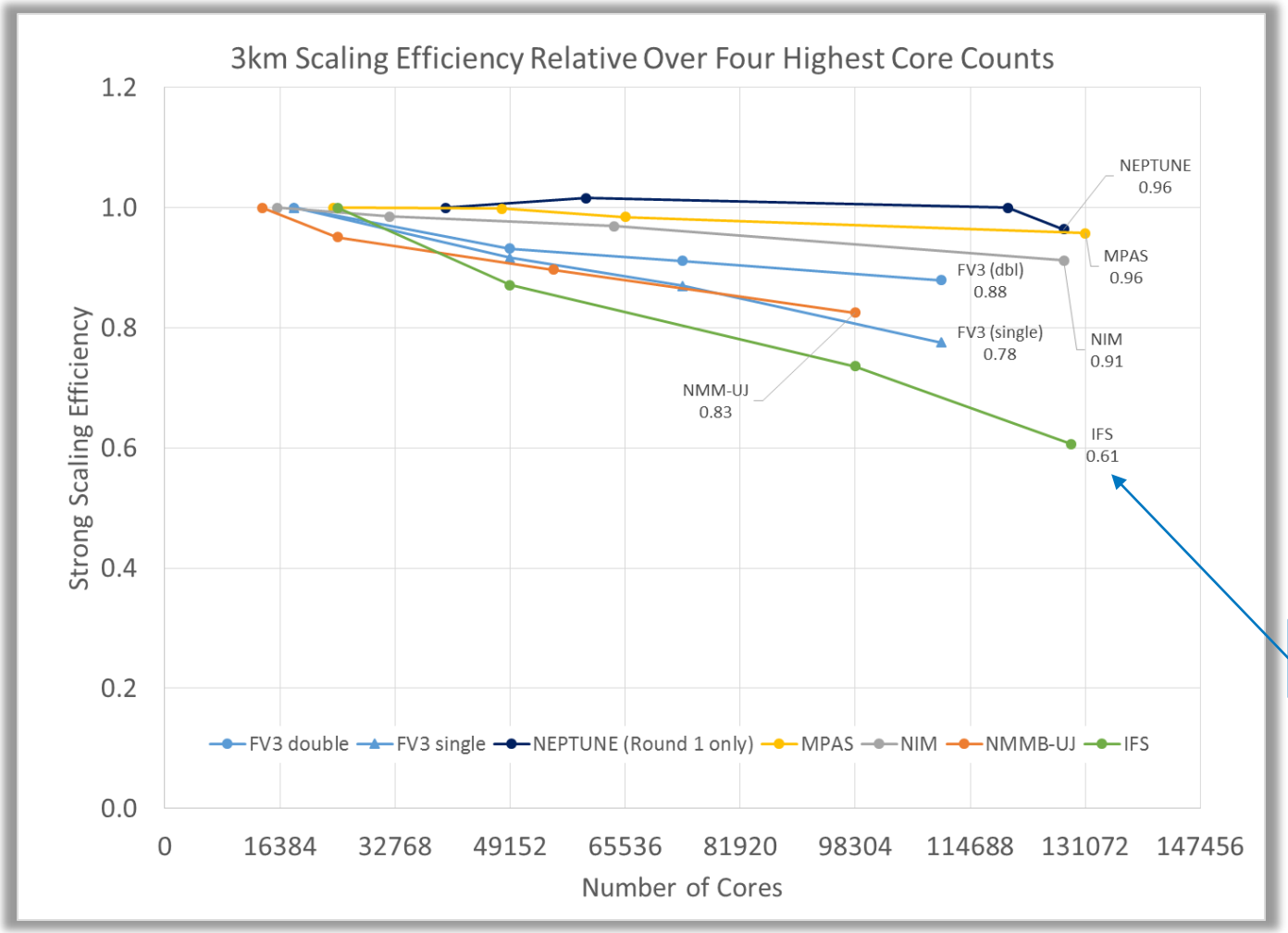
- Performance:
  - Number of processor cores needed to meet operational speed requirement with 13-km workload
  - Rankings (fastest to slowest): NMM-UJ, FV3, NIM, MPAS, NEPTUNE



ECMWF Guest Dycore

AVEC Level-1 Evaluations: Scalability

- Scalability: ability to efficiently use large numbers of processor cores
  - All codes showed good scaling.
  - Rankings (most to least scalable): NEPTUNE, MPAS, NIM, FV3, NMM-UJ



ECMWF Guest Dycore

AVEC Level-1 Evaluations: Software

Software evaluations intended to highlight strengths and weak-nesses of codes to be ready for NGGPS

Note: snapshot in time, all codes under active development

Additional evaluation including detailed code inspection and review of documentation will continue into Level-2 testing

	FV3	NMMUJ	MPAS	NIM	NEPTUNE
Languages	Fortran, Co-Array Fortran, and C	Fortran	Fortran and C	Fortran	Fortran and C
MPI	2D horz. decomposition and over cube faces; Nearest neighbor	2D horz. decomposition and over cube faces; Nearest neighbor	2D horz. decomposition; Nearest neighbor	2D horz. decomposition; Nearest neighbor; Specified with SMS (ESRL tool) directives	2D horz. decomposition; Nearest neighbor
OpenMP	Loop-level, threaded dimension varies; subroutine level around physics	Mostly loop-level, threaded dimension varies; Subroutine level over radiation	Planned (1-2 years); Subroutine level	Loop level, threading is consistently over horizontal dimension; subroutine level around physics	Planned
Storage/Loop Nesting Order	IJK	IJK	K-innermost	K-innermost	K-innermost
Extensible software design	Abstraction layers and APIs for domain definition & mgmt., comms, I/O, etc.; Supports Init/Run/Finalize component interfaces; Uses FMS Framework	Use of Fortran MODULES; Flat program structure (everthing called from main); NEMS/ESMF (planned)	Abstraction layers and APIs for domain definition & mgmt., comms, I/O, etc.; Supports Init/Run/Finalize component interfaces	Abstraction layers and APIs for domain definition & mgmt., comms, I/O, etc.; Supports Init/Run/Finalize component interfaces	Use of Fortran MODULES (I/O, domain decomposition, physics, dynamics right hand side, dynamics time integration, etc.). Flat program structure.
Nesting/Refine ment	Nesting within a cubed-sphere face or stretched grid; 1-way and 2-way; static (non-moving); plans for moving nests	Plans for moving 1- and 2-way interacting nests; no geographic restrictions on movement	In-place refinement (inherently 2-way); static (non-moving) refinement	Plans for 1- and 2-way interacting nests; static (non-moving)	In-place refinement (inherently 2-way); adaptive mesh refinement planned; no cost estimate
Coding practices	IMPLICIT NONE; INTENT(IN/OUT/INOUT); Argument keywords used; Kalnay-conforming physics APIs	IMPLICIT NONE; INTENT(IN/OUT/INOUT); Some argument keywords; Kalnay-conforming physics APIs	IMPLICIT NONE; INTENT(IN/OUT/INOUT); Some argument keywords	IMPLICIT NONE; INTENT(IN/OUT/INOUT); Argument keywords used; Kalnay-conforming physics APIs	IMPLICIT NONE; INTENT(IN/OUT/INOUT); Some argument keywords
Reproduciblity	Bit-for-bit reproducible on different core counts; Bit-for-bit restarts	Bit-for-bit reproducible on different core counts; Bit-for-bit restarts (planned)	Not bit-reproducible on different numbers of MPI tasks (but under development); Bit-for-bit restarts	Bit-for-bit reproducible on different core counts; Bit-for-bit restarts	Not bit-reproducible on different numbers of MPI tasks; Bit-for-bit restarts
Current and Advanced Architectures	Originally developed for vector systems; has adapted to each new architectural paradigm and now developed and supported on conventional multi-core processors. Hybrid MPI/OpenMP makes well suited for MIC; exploring GPU with directives-based or CUDA programming models.	Developed and supported on conventional multi-core processors; RRTMG physics has been adapted to accelerators (MIC and GPU).	Developed and supported on conventional multi-core processors. No efforts currently for novel architectures.	Developed and supported as single-source on conventional multi-core, MIC and GPU.	Developed and supported on conventional multi-core processors. Testing kernels on MIC and GPU.