Meeting beam physics grand challenges via HPC

Presenters: Jean-Luc Vay (LBNL) & David Bruhwiler (RadiaSoft)

GARD ABP Workshop 2 – WG2

April 22-23, 2020 – Organized by FNAL – via Zoom

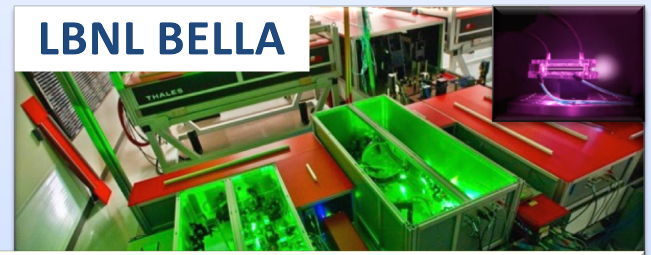
Other Berkeley Lab contributors: A. Huebl, R. Lehe, C. Mitchell, G. Penn, J. Qiang, R. Ryne, M. Thévenet

Other RadiaSoft contributors: N. Cook, D.T. Abell, E. Carlin, J. Edelen, C. Hall,
M. Keilman, P. Moeller, R. Nagler, B. Nash and S. Webb

This work is supported by the US DOE, Office of Science, Office of High Energy Physics, Office of Nuclear Physics, Office of Basic Energy Sciences and Office of Advanced Scientific Computing Research.

All accelerators in the world rely on modeling

and increasingly on *high-performance computing*



Next generation of accelerators needs next generation of HPC modeling tools!

Outline

- EVA – E*nd-to-End Virtual Accelerator*
    - *Required infrastructure*
    - *Need for best practice and community development*
- How can it fail?
- Connection to grand challenges
- Who is working on this Now? Collaborations?
- 5+ year development roadmap

EVA – End-to-end Virtual Accelerator

Predict behavior of beams in accelerators "as designed/built"

Fast – *runs in seconds to minutes*

Hi-Fi – *full & accurate physics*

Link – *integrated ecosystem*

Vision

Real-time virtual prototyping of entire accelerators

Like **flight simulator**: ***EVA*** to allow to model operation of the accelerator in "real time".

Synergies

U.S. DEPARTMENT OF ENERGY | Office of Science

Fast	ASCR
Hi-Fi	HEP + BES + NP + FES
Link	CAMPA Consortium for Advanced Modeling of Particle Accelerators

Support from the DOE SBIR program: HEP + BES + NP + ASCR

Can leverage large investments from ASCR.

SciDAC Scientific Discovery through Advanced Computing

ECP EXASCALE COMPUTING PROJECT

Standardization to couple codes & uniformize data for machine learning.

PICMI Standard

openPMD

Online framework for integrated, collaborative research and education.

Sirepo

Infrastructure required to support "Virtual Accelerators"

- The proposed R&D will enable
  - Code interoperability
    - Rapid benchmarking between many different codes
    - Code coupling (e.g. IMPACT / ELEGANT / MAD-X / MARS)
    - Modular physics – rapid interchange from low-dimensionality / reduced models to full physics
  - Computational reproducibility
    - Archive key simulations (design process) for use years later (e.g. commissioning)
    - Instantaneous collaboration (distributed teams; sharing dynamic results with leadership)
  - Accessibility (cloud computing, documentation, graphical user interfaces)
    - Scientific leaders and experimentalists need direct access to the simulations
    - User Facility approach to computing (assist outsiders with state-of-the-art codes)
    - Integration of codes with facilities, including direct interaction with the control system
- Present state of the art:
  - complicated command-line workflows; each lab or group has its own code(s)

Support of best practices and community development are essential

- Support development - & encourage usage - of standardized libraries (e.g., Kokkos, RAJA, CABANA, AMReX) to minimize overhead of new code development & duplication.

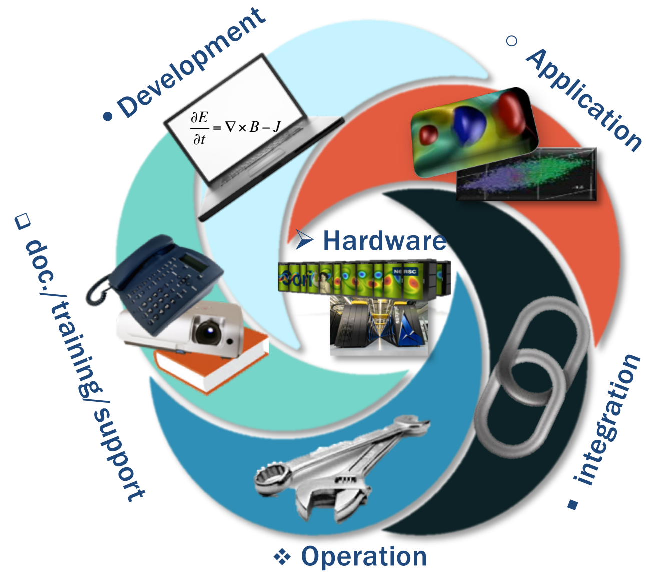
- Increase codes reliability: encourage validation, verification, version control, open source.

- Support common standards for output data & input scripts for interoperability & feed AI/ML engines.

- Support for code documentation, training, user support.

- Support for software engineers to develop, maintain & port codes to new architectures.
  - Complexity of development on new platforms (e.g., GPUs) is increasing ➔ needs specialists.

- Support for large scale capacity (ensembles) & capability (big runs) computing on clouds and supercomputers.



Outline

- EVA – E*nd-to-End Virtual Accelerator*
  - *Required infrastructure*
  - *Need for best practice and community development*
- How can it fail?
- Connection to grand challenges
- Who is working on this Now? Collaborations?
- 5+ year development roadmap

How can it fail?

- Business as usual
  - separate codes
  - developed by physicists without inclusion of, software engineers, computer scientists and applied mathematicians
  - no adoption of standards for input scripts and output data
  - no transition to modern architectures and languages (the industry standards evolved)
    - GPUs as well as new CPUs (PPC64le, ARM, RISC-V, ...)
    - Language support: C++ is the major language, Fortran compilers lag behind
  - Irreproducible results:
    - not fully open codes (not openly distributed, including source; not open to community contributions)
    - cross-community view: e.g., Supercomputing meeting series strongly encourages papers with code source
  - insufficiently documented codes

➔ Success will require us to work together

- Lack of development of the required workforce
- Lack of funding

Outline

- EVA – E*nd-to-End Virtual Accelerator*
    - *Required infrastructure*
    - *Need for best practice and community development*
- How can it fail?
- Connection to grand challenges
- Who is working on this Now? Collaborations?
- 5+ year development roadmap

All four Grand Challenges connect to computing

GC #1. Increase beam intensities by orders of magnitude.

➔ increase # of particles by orders of magnitude.

GC #2. Increase beam phase-space density by orders of magnitude, towards quantum degeneracy limit.

➔ increase precision of simulations; add quantum effects.

GC #3. Control the beam distribution down to the level of individual particles.

➔ simulate all the particles.

GC #4. Develop predictive “virtual particle accelerators”.

➔ simulate everything: all the particles, conductors, dark currents, many turns, …

Need for fast, multiphysics codes on supercomputers/clouds.

Outline

- EVA – E*nd-to-End Virtual Accelerator*
  - *Required infrastructure*
  - *Need for best practice and community development*
- How can it fail?
- Connection to grand challenges
- Who is working on this Now? Collaborations?
- 5+ year development roadmap

Who is working on this now? Collaborations?
Many people/teams – need for community effort

- Many accelerator and beam physics codes
  - Mostly single developers; a few teams, some large.

Who is working on this now? Collaborations?
Many people/teams – need for community effort

- **Many accelerator and beam physics codes**
  - Mostly single developers; a few teams, some large.

Beam dynamics codes section from Accelerator Handbook (A. Chao, 2013)

Duplication?

Beam Dynamics Codes:

(Below, PIC refers to codes with particle-in-cell space-charge capability.)

Code	URL or Contact	Description/Comments
ASTRA	tesla.desy.de/~meykopff	3D parallel, general charged particle beams incl. space charge
AT	sourceforge.net/projects/atcollab/	Accelerator Toolbox
BETACOOL	betacool.jinr.ru	Long term beam dynamics: ECOOL, IBS, internal target
Bmad, Tao	www.lns.cornell.edu/~dcs/bmad/	General purpose toolbox library + driver program
COSY INFINITY	www.cosyinfinity.org	Arbitrary-order beam optics code
CSRTrack	www.desy.de/xfel-beam/csrtrack	3D parallel PIC; includes CSR; mainly for e^- dynamics
Elegant/SDDS suite	aps.anl.gov/elegant.html	parallel; track, optimize; errors; wakes; CSR
ESME	www-ap.fnal.gov/ESME	Longitudinal tracking in rings
HOMDYN	Massimo.Ferrario@LNF.INFN.IT	Envelope equations, analytic space charge and wake fields
IMPACT code suite	amac.lbl.gov	3D parallel multi-charge PIC for linacs and rings
LAACG code suite	laacg.lanl.gov	Includes PARMILA, PARMELA, PARMTEQ, TRACE2D/3D
LiTrack	www.slac.stanford.edu/~emma/	Longitudinal linac dynamics; wakes; GUI-based; error studies
LOCO	safranek@slac.stanford.edu	Analysis of optics of storage rings; runs under matlab
LUCRETIA	www.slac.stanford.edu/accel/ilc/codes	Matlab-based toolbox for simulation of single-pass e^- systems
MaryLie	www.physics.umd.edu/dsat	Lie algebraic code for maps, orbits, moments, fitting, analysis
MaryLie/IMPACT	amac.lbl.gov	3D parallel PIC; MaryLie optics + IMPACT space charge
MAD-X	mad.web.cern.ch/mad	General purpose beam optics
MERLIN	www.desy.de/~merlin	C++ class library for charged particle accelerator simulation
OPAL	amas.web.psi.ch	3D parallel PIC; cyclotrons, FFAGs, linacs; particle-matter int.
ORBIT	jzh@ornl.gov	Collective beam dynamics in rings and transport lines
PATH	Alessandra.Lombardi@cern.ch	3D PIC; linacs and transfer lines; matching and error studies
SAD	acc-physics.kek.jp/SAD/sad.html	Design, simulation, online modeling & control
SIMBAD	agsrhichome.bnl.gov/People/luccio	3D parallel PIC; mainly for hadron synchrotrons, storage rings
SIXTRACK	frs.home.cern.ch/frs/	Single particle optics; long term tracking in LHC
STRUCT	www-ap.fnal.gov/users/drozhdin	Long term tracking w/ emphasis on collimators
Synergia	https://compacc.fnal.gov/projects	3d parallel PIC: space charge, nonlinear tracking and wakes
TESLA	lyyang@bnl.gov	Parallel; tracking; analysis; optimization
TRACK	www.phy.anl.gov/atlas/TRACK	3D parallel PIC, mainly for ion or electron linacs
LIBTRACY	libtracy.sourceforge.net/	Library for beam dynamics simulation
TREDI	www.tredi.enea.it	3D parallel PIC; point-to-point Lienard-Wiechert
UAL	code.google.com/p/ual/	Unified Accelerator Libraries
WARP	DPGrote@lbl.gov	3D parallel ES and EM PIC with accelerator models
ZGOUBI	sourceforge.net/projects/zgoubi/	Magnetic optics; spin; sync radiation; in-flight decay

Who is working on this now? Collaborations?

Many people/teams – need for community effort

- **Many accelerator and beam physics codes**
  - Mostly single developers; a few teams, some large.

Plasma accelerator codes
From J.-L. Vay, R. Lehe, *RAST 9* (2016)

Definitely some duplication.

Table 1. List of simulation PIC codes for the modeling of plasma accelerators.

Code	Type	Website/reference	Availability/license
ALaDyn/PICCANTE	EM-PIC 3D	http://aladyn.github.io/piccante	Open/GPLv3+
Architect	EM-PIC RZ	https://github.com/albz/Architect	Open/GPL
Calder	EM-PIC 3D	http://iopscience.iop.org/article/10.1088/0029-5515/43/7/317	Collaborators/Proprietary
Calder-Circ	EM-PIC RZ^+	http://dx.doi.org/10.1016/j.jcp.2008.11.017	Upon Request/Proprietary
CHIMERA	EM-PIC RZ+	https://github.com/hightower8083/chimera	Open/GPLv3
ELMIS	EM-PIC 3D	http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A681092&dswid=-8610	Collaborators/Proprietary
EPOCH	EM-PIC 3D	http://www.ccpp.ac.uk/codes.html	Collaborators/GPL
FBPIC	EM-PIC RZ^+	https://fbpic.github.io	Open/modified BSD
HiPACE	QS-PIC 3D	http://dx.doi.org/10.1088/0741-3335/56/8/084012	Collaborators/Proprietary
INF&RNO	QS/EM-PIC RZ	http://dx.doi.org/10.1063/1.3520323	Collaborators/Proprietary
LCODE	QS-PIC RZ	http://www.inp.nsk.su/~lotov/lcode	Open/None
LSP	EM-PIC 3D/RZ	http://www.lspsuite.com/LSP/index.html	Commercial/Proprietary
MAGIC	EM-PIC 3D	http://www.mrcwdc.com/magic/index.html	Commercial/Proprietary
Osiris	EM-PIC 3D/RZ^+	http://picksc.idre.ucla.edu/software/software-production-codes/osiris	Collaborators/Proprietary
PHOTON-PLASMA	EM-PIC 3D	https://bitbucket.org/thaugboelle/ppcode	Open/GPLv2
PICADOR	EM-PIC 3D	http://hpc-education.unn.ru/en/research/overview/laser-plasma	Collaborators/Proprietary
PIConGPU	EM-PIC 3D	http://picongpu.hzdr.de	Open/GPLv3+
PICLS	EM-PIC 3D	http://dx.doi.org/10.1016/j.jcp.2008.03.043	Collaborators/Proprietary
PSC	EM-PIC 3D	http://www.sciencedirect.com/science/article/pii/S0021999116301413	Open/GPLv3
QuickPIC	QS-PIC 3D	http://picksc.idre.ucla.edu/software/software-production-codes/quickpic	Collaborators/Proprietary
REMP	EM-PIC 3D	http://dx.doi.org/10.1016/S0010-4655(00)00228-9	Collaborators/Proprietary
Smilei	EM-PIC 2D	http://www.maisondelasimulation.fr/projects/Smilei/html/licence.html	Open/CeCILL
TurboWave	EM-PIC 3D/RZ	http://dx.doi.org/10.1109/27.893300	Collaborators/Proprietary
UPIC-EMMA	EM-PIC 3D	http://picksc.idre.ucla.edu/software/software-production-codes/upic-emma	Collaborators/Proprietary
VLPL	EM/QS-PIC 3D	http://www.tp1.hhu.de/~pukhov/	Collaborators/Proprietary
VPIC	EM-PIC 3D	http://github.com/losalamos/vpic	Open/BSD clause-3 license
VSim (Vorpal)	EM-PIC 3D	https://txcorp.com/vsim	Commercial/Proprietary
Wake	QS-PIC RZ	http://dx.doi.org/10.1063/1.872134	Collaborators/Proprietary
Warp	EM-PIC 3D/RZ^+	http://warp.lbl.gov	Open/modified BSD

EM = electromagnetic; QS = quasistatic; PIC = particle-in-cell; 3D = three-dimensional; RZ = axisymmetric; RZ^+ = axisymmetric with azimuthal Fourier decomposition.

Who is working on this now? Collaborations?
Many people/teams – need for community effort

- Many accelerator and beam physics codes
  - Mostly single developers; a few teams, some large.
  - Some teams benefit(ted) from DOE ASCR Supercomputing projects:
    - SciDAC 1-3: ComPASS (ACE3P, Impact, Osiris, QuickPIC, Synergia, Warp).
    - SciDAC 4: ComPASS (Synergia, QuickPIC).
    - Exascale Computing Project: WarpX.

> Lots of time & efforts went into developing those codes.
> Should capitalize and redirect toward community framework(s).
> ➔ needs team work!

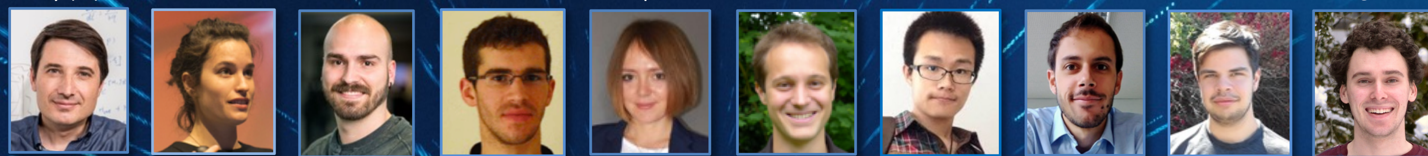
(examples next slides)

WarpX team*: physicists + applied mathematicians + computer scientists

ECP

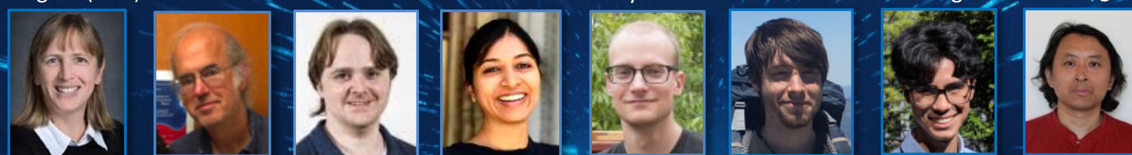
BERKELEY LAB

Jean-Luc Vay (PI) | Diana Amorim | Axel Huebl | Rémi Lehe | Olga Shapoval | Maxence Thévenet | Yinjian Zhao | Edoardo Zoni | Glenn Richardson | Daniel Belkin



ATAP
ACCELERATOR TECHNOLOGY & APPLIED PHYSICS DIVISION

Ann Almgren (coPI) | John Bell | Kevin Gott | Revathi Jambunathan | Andrew Myers | Michael Rowan | Eloise Yang | Weiqun Zhang



Kevin Gott (NESAP)

Michael Rowan (NESAP)

CRD AMReX

NERSC NESAP

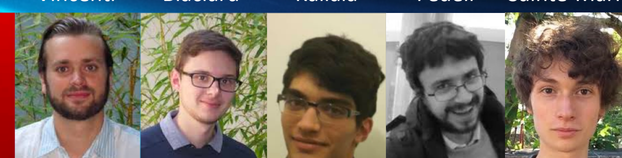
David Grote (coPI)

Marc Hogan (coPI) | Lixin Ge | Cho Ng

SLAC

Henri Vincenti | Guillaume Blaclard | Haithem Kallala | Luca Fedeli | Antonin Sainte-Marie

cea PARIS-SACLAY



+ collaborators from CEA Saclay (France)

The project also leverages other ASCR (ECP & others) efforts via adoptions of other tools/methods, often via collaboration.

*Many at fraction of time on WarpX.

Sirepo team:* software engineers + physicists + machine learning + writer + product mgmt

*Many at a small fraction of their time on Sirepo.

- Software Engineering
  - cloud computing
  - browser-based GUIs
- Physics & Data Science
  - machine learning
  - particle accelerators
  - plasma devices
  - control systems
  - COMSOL Multiphysics
- Product Development
  - product management
  - science writing

Rob Nagler

Paul Moeller

Evan Carlin

Mike Keilman

Sirepo

sirepo.com

https://github.com/radiasoft/sirepo

Xi Tan

Dan Abell

David Bruhwiler

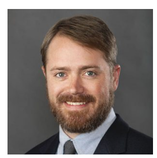
Johan Carlsson

Nathan Cook

Chris Hall

Ilya Pogorelov

Stephen Coleman



Nick Goldring

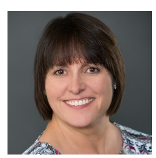
Callie Federer

Jon Edelen

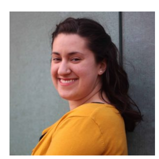
Stephen Webb

Boaz Nash

Paula Messamer



Emily “Remy” Poore



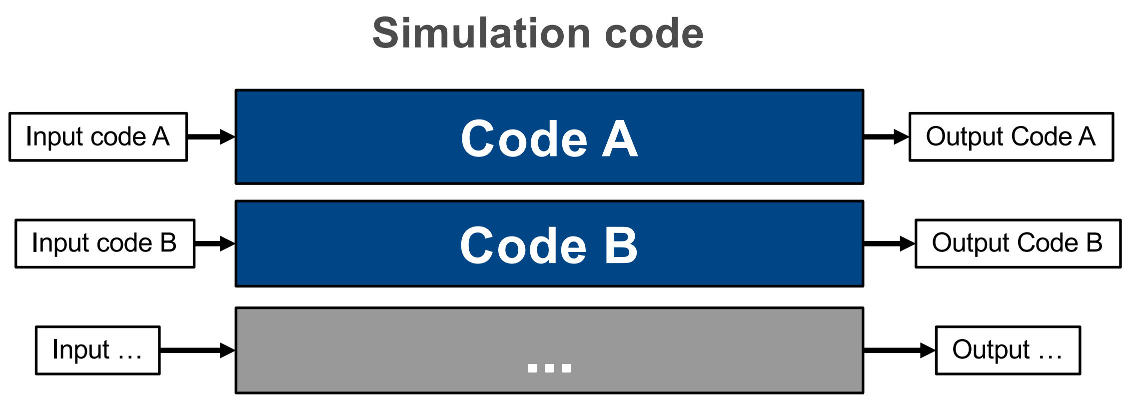
radiasoft

Who is working on this now? Collaborations?

Emergence of standards for inputs/outputs

No standards

Different input formats



Different output formats

Difficult for benchmarks, workflows, optimization or AI/ML engines, ...

Fun comparison: imagine the car makers not following standards and having to learn from scratch to drive any new car or rental...

Who is working on this now? Collaborations?

Emergence of standards for inputs/outputs

With standards

Particle-In-Cell Modeling Interface

PICMI Standard

→

Simulation code

Code A

Code B

...

→

open Particle Mesh Data standard

openPMD

New extension by David Sagan and Christopher Mayes

Consortium for Advanced Modeling of Particle Accelerators CAMPA

DOE HEP Gard

Facilitate:

- Chaining of codes for multiphysics workflow.
- Cross-benchmarking, verification, comparison.
- Interfacing with ensemble optimization, AI/ML software.
- Integration into a framework.

Who is working on this now? Collaborations?
Solutions for ABP computing framework have emerged

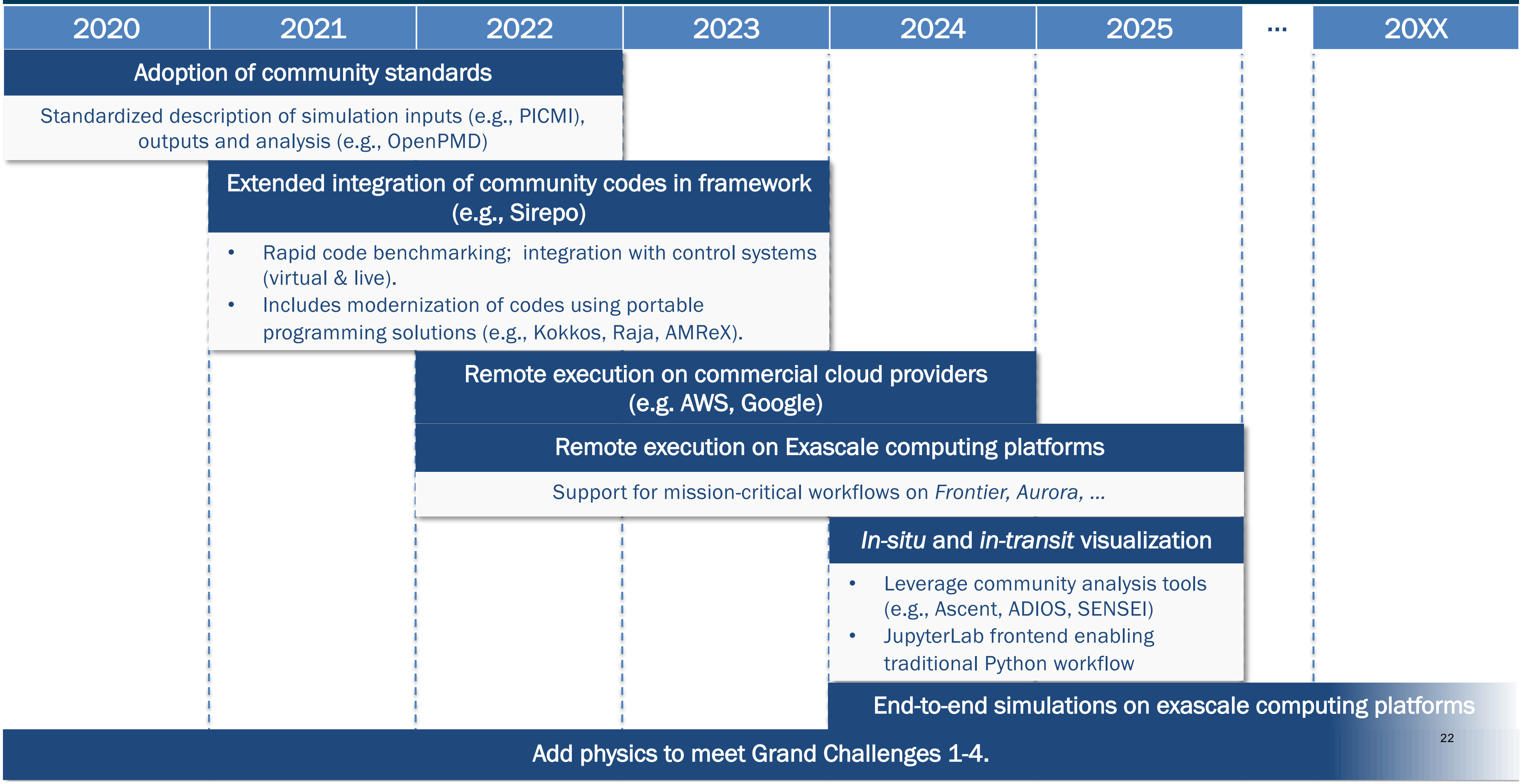
- Required characteristics of an ABP computing framework
  - Open Source, with public development (e.g. GitHub, Bitbucket, Gitlab…)
  - Supports many codes, with no requirements imposed on the code developers
    - e.g. Sirepo supports: Synergia, Opal, Warp*, MAD-X***, elegant, Zgoubi, SRW, Shadow, Radia**
    - must also support reduced models, code coupling & code interchange
  - Provides ease of use for students and other non-experts
    - e.g. Sirepo is used regularly for USPAS, Korea PAS, graduate courses in X-ray optics.
  - Supports relevant facilities and particle accelerator technologies
    - e.g. hadron accelerator chain (PIP II), plasma-based (BELLA, FACET-II), light sources
  - Reproducible computing
    - Technology for archive/recovery of code & full dependencies; <u>can be rerun years later</u>
    - e.g. this capability is ready for beta testing in Sirepo
  - Must enable use of all hardware platforms, without locking users in
    - User-friendly solution for simulations on laptops, desktop, supercomputers and cloud
    - Easy interchange between “expert mode” and user-friendly mode

*partial support **in development **future plans (funded)

Outline

- EVA – E*nd-to-End Virtual Accelerator*
    - *Required infrastructure*
    - *Need for best practice and community development*
- How can it fail?
- Connection to grand challenges
- Who is working on this Now? Collaborations?
- 5+ year development roadmap

5+ Year Development Roadmap (strawman)



Thanks for your attention.

Comments? Questions?