##

Your comments to the CompF Report

##

Please enter your comment about the CompF report in the following. Please specify the section and line numbers, and possibly (recommended) the author of the comment and email address. We will strive to address all the comments we receive.

Template:
Comment by: *Name, email address*
Date:
Report Version no.:
Section Number: XYZ
Line Numbers: *X Y Z*
Comment text

Comment by: Krzysztof Genser@fnal.gov
Date: 2022 07 22
Report Version no.: the summary slides, recommendations

Thanks for providing the recommendations.
Suggestions to slightly rephrase 1 & 3

1: The US HEP community should take a leading role in continuous development, maintenance, and user support of software packages essential to US based groups.

3: Support computing professionals/researchers, and physicists for code re-engineering and adaptation to use heterogeneous resources effectively. To serve the needs of hard to parallelize algorithms, while attempting to rewrite them, traditional CPU-based hardware should coexist with heterogeneous resources.

Comment by: Stefan Roiser, stefan.roiser@cern.ch

Date 18 july 2022
Report Version no: v0 (?)
Section Number: 3.4 Physics generators
Line Numbers: 428 ff.

In the context of speeding up event generators it may be worth mentioning something following along the lines below

Checking event generators (e.g. Madgraph5_aMC@NLO) reveals that the bulk of the CPU time is spent in the matrix element calculation and increases even more with the complexity of the underlying physics process. Parallelising the matrix element calculations on the event level can be done in lockstep, i.e. the parallel processing of N events starts and ends at the same moment in time, which is an excellent match for performance improvements via hardware accelerations on GPUs and through vector instructions on CPUs. Investigations in this direction for various generator packages should be envisaged.

A few references in case

- HSFWS2020: https://indico.cern.ch/event/941278/contributions/4101793/
- vCHEP2021: https://doi.org/10.1051/epjconf/202125103045
- ICHEP2022: https://agenda.infn.it/event/28874/abstracts/20368/

Comment by: Krzysztof Genser@fnal.gov
Date: 2022 07 19
Report Version no.: v0
Executive Summary
Thanks for including Geant4 in it :)
Perhaps add a recommendation sentence for each of the identified critical challenges? Although, it seems the main recommendation is to stand the committee, so it may be a task for that body.
Section Number: 3.4
Line Numbers: 373: perhaps add: "muon, " (experiments)
376: perhaps add: "potentially "(computationally expensive)
Regarding Bibliography:
line: 940
Geant4 references should be:
Recent Developments in Geant4, J. Allison et al., Nucl. Instrum. Meth. A 835 (2016) 186-225

Geant4 Developments and Applications, J. Allison et al., IEEE Trans. Nucl. Sci. 53 (2006) 270-278
Geant4 - A Simulation Toolkit, S. Agostinelli et al., Nucl. Instrum. Meth. A 506 (2003) 250-303
See e.g., CompF2 bibliography section
line 1008: add "In 2022 Snowmass Summer Study,"

Comment by: *Ianna Osborne, ianna.osborne@cern.ch*
Date: 07.19.2022
Report Version no.:
Section Number: 6
Line Numbers: *733, 734, 747, 758*

Line 733: “software ecosystem”
Line 734: There are two major ecosystems… The ROOT suit… and Python

This sounds a bit odd to me. Perhaps, “two major players” ? Usually software ecosystems are defined as “a collection of **software projects** that are developed and co-developed in the same environment”. In order to describe an analysis that takes into account multiple software systems, ecosystem metaphors are used. The term “Python ecosystem” is widely used and understood.

Line 747: “feature junior personnel”

Thanks, it is flattering :-)

Line 758: Python “is not necessarily an optimal language for scientific computing”

It’s an arguable assumption. Python on its own can be slow, but it binds well to C++ and specifically ROOT. The GIT compiled Python functions are as fast as C++.

Comment by: Zach Marshall, ZLMarshall@lbl.gov
Date:18 July 2022
Report Version no.: I don’t see any report version number in the PDF

On the executive summary:

“theoretical calculations”

In the second item (cross-cutting developments), it might be nice to include what we at the LHC refer to as “Production System” software. These are tools like Rucio, Panda, Dirac, etc that manage distributed data or distributed workloads on disjoint sites, and of course are very effective when shared across multiple experiments (those three examples all are).

I think in point 3, “runs on a particular machine” is a little too vague even for an executive summary. We shouldn’t give the impression that if we run at Fermilab we can’t run at CERN. It’s also true that a great many groups are well-equipped to use HPC and Cloud resources, even if they are not well equipped to use e.g. accelerators that might be available with those resources.

I think point 4 should be rephrased a bit. We shouldn’t suggest that a physics department should hire a “Software and Computing researcher” – but that’s not what many of these people are. They are *physicists* who are experts in software and computing aspects of their field. Similarly, it’s not that we need staff positions for S&C researchers, it’s that we need staff and faculty positions for physicists who are S&C experts.

General / structural / more important comments:

The authorship of the document seems to be dominated by collider physicists. It would be useful to go back and consider whether the examples given in various places (and the definitions given) are relevant also to the other communities. For example, frequent mention is made of “the grid”, which is not a single universal entity, nor an entity with which all experiments engage.

The style of the document varies significantly by section, with e.g. Section 2’s and 3’s conclusions relying on significant bolding; Section 5 makes heavy reference to the input documents and their specific sections, in contrast to the other sections. It would be nice to harmonize these style choices.

The term “heterogeneous computing” is used frequently without a clear meaning. I know of no software package that benefits from heterogenous computing. I know of many that benefit from accelerators of one kind or another. A homogeneous GPU+CPU system might be highly beneficial – but that is homogeneous, not heterogeneous. An FPGA system might benefit some software – but again, that is homogeneous. If when “heterogeneous computing” is written, what is meant is “the availability of non-x86 hardware”, then this should be stated clearly somewhere (and I think a different term should be used, but that’s a different objection). Often it seems that what is meant is “a farm with a large number of GPUs.”

There are a number of places (I try to point out a few below) where rather bold claims are made without a reference. It would be helpful to go back and ensure it is clear what supports these claims.

It is surprising that there is no mention of ARM or SoC systems in the document.

It's also a bit surprising that the only mention of HEP-CCE is at L678, and the only mentions of IRIS-HEP a few lines later (no mention of IRIS-HEP in the analysis section?).

L50 – This point about training comes a bit out of the blue. The assertion seems to be that because we have many approaches to accelerators, training is a challenge? Should this imply that CPU-only software development is now straightforward? I would say it is not. This should be clarified and rephrased.

L101 – Reference 15 absolutely does not support this claim, which should be removed. First, it is an analysis of CMS user repositories on GitHub, which may or may not even be representative of the LHC, to say nothing of the broader HEP community. Second, the statement is about fractions of repositories using such-and-such a language. It doesn't distinguish in any way between people wanting to make a pretty plot in python, or people doing an entire complex analysis chain in python.

L263 – Do you really want to advocate a faculty tenure track appointment in "software and computing"? Why is this not an appointment for a physicist who has expertise in software and computing?

L367 – The reference is talking about computing up to 2016, which is a far cry from "during the LHC Run 2". Moreover, the number reported here for LHCb ("up to 90% for LHCb") is a projection to 2020 mentioned in the reference, _not_ an indication of what has actually happened. This should be corrected. Why not mention ALICE here as well?

L380-1 – About "adapting to the HPC environment": this doesn't make sense. If a system is an HTC problem, one should not try to twist it into requiring additional inter-node communication (the biggest difference between HTC and HPC). If this is meant to be about use of accelerators, then it confuses "HPC" and "Accelerators" (accelerators can of course be in HTC centers, and HPC centers can exist without accelerators!). This distinction is made nicely around L790-794.

L521 – This section begins describing areas of machine learning that require dedicated solutions, and I think does reasonably well there (up to L529). After that seem to come a list of areas of application where it's unclear whether there is any additional ML development required – with one mention of work to cast HEP problems as ML problems that can be solved with existing ML tools (perhaps this is on the boundary). This section seems to call to be divided up or recast in a different light.

L604 – “These computers all feature GPUs” – this is not true at any scale and should be removed. Fugaku is a very nice example of a large, non-GPU HPC.

L611 – It might be nice here to point out that the experiments have attempted to do so, and to cite a few of the places where these projections are provided.

L660 – “but at what cost” – this should be either removed or unpacked.

L666 – Why is this benchmarking suite only for AI/ML applications? If it’s not, then why isn’t some reference to HEPSCORE made here?

Section 5 seems to be very LHC-centric; should more be said about other subfields?

Section 5 is, I believe, the only one that does not mention training or support for personnel. Is training required in this area? Is additional support for personnel? Is this difference intentional or accidental?

Minor / Textual / Editorial Comments:

The references need work. Many are incomprehensible. I don’t know why we’d provide a month and not an arXiv number, for example.

L38 – These are even specific *versions* of popular software programs (now Pythia8 will be cited, and Geant3 would have been cited some years ago).

L40 – It’s really unclear what is meant by “computing limitations are on par with detector limitations.” Measured how?

L47 – “The issue is not one for which there is a single approach”? This is a rather convoluted sentence; I suggest it be rephrased.

L51 – What does “over the next 10 years with a 20 year horizon” mean? Either you’re examining it out 10 years, or you’re looking out 20 years, not both.

L55 – The HL-LHC is scheduled to run until 2041. It also is odd to talk about the HL-LHC producing more data “than the LHC experiments”. The HL-LHC won’t produce any data. ATLAS and CMS will, and they are the same experiments, upgraded, that will exist in the HL-LHC. Would one talk about Tevatron’s Run 2 as having produced “more data than CDF and D0”? Also, this “order of magnitude more” line is close (a bit exaggerated) for ATLAS and CMS, but not for the other experiments?

L56 – we should be careful with the term “edge”, as it also has special meaning for HPC systems. In fact, you use it in the other sense down around L644-645, for example, and again

around L696. Why not refer to the TDAQ white papers here? Perhaps “Low-latency” and “High-latency” is the point?

Fig 0-1 – If you use high precision on the right for Dune, then the precision for the HL-LHC could be improved. I believe projections for ATLAS and CMS event sizes are around 4 MB/event, and the collision rate around 30 MHz (the ring is not full), which brings the data rate up to 120 TB/s. Does the DUNE 400 PB number include simulations? I think for the HL-LHC the projections are O(5 EB) of data and simulation per experiment – but of course the numbers are quite different for ATLAS and CMS compared to LHCb or ALICE. In the caption, are these numbers really “not meant to be illustrative”?

L87 – I would suggest to remove “Access” from the bold naming of this area. Access suggests that this is really about access and authentication, while the group had a much broader mandate than that. In fact, it looks like the issues of access and authentication don’t even get a mention in the introduction.

L96 – What is "the HEP grid”? Do you mean the WLCG? OSG? Some loose combination of all computing sites on earth? Anything connected to the internet?

L148 – Missing reference in “Sec .".

L155 – It would be useful to acknowledge here that a flat budget does not mean constant storage or compute resources. They will grow.

L170 – It’s unclear to me what point this paragraph is trying to make about frameworks, unless it’s that “they are hard"; I would say that all of the requirements described in this section are satisfied by Ray, for example, so we should not leave the impression that it is up to our community to solve this problem or that we are the only ones affected by it.

L181-2 – This seems to suggest that collider physics is “easy”, while cosmology is “not easy”. It also seems to suggest that signal is clear in collider physics, and obscure in cosmology. I don’t think either is fair. I think I understand what you’re trying to say, but better to write it clearly.

L203 – missing “A” in “AI/ML”.

L223-236 – this seems like it should be made a bulleted list.

L245 – This screams for a reference to support these claims.

L254 – Why specifically GPUs here?

L338 – Any reference to cite to support these claims?

L345 – “Additionally, the workload needs” (missing “the”)

L348 – “accelerator simulation runs efficiently” (I think just remove “they”?)

L391 – I suppose I understand why MLaaS is mentioned in this white paper, but this placement doesn’t make sense. For something like detector simulation, which is one of the major processing loads of any experiment, you would not want to use an off-node service for calculations that are being done constantly. MLaaS helps for rare workloads, where purchasing a GPU doesn’t make sense. Otherwise, what we’re really talking about is scheduling across diverse compute nodes in a single center (some with GPUs and some without), which is a different problem. I suggest moving this sentence to elsewhere in the document. In fact, this is mentioned down towards L669, so this could be dropped or moved there.

L441 – “to future experiments” (plural)

L495 – Missing reference

L569 – I don’t think “local” has anything to do with it. Rather “dedicated” might have a lot to do with it. It’s irrelevant whether the GPU is physically under my desk or is in a computing center a few states away, as long as I can play with it as I please, right?

L600+ – While I think the reliance on references is good here, this is so heavy that it makes the conclusions difficult to extract.

L663-5 – while Real-time analysis is clearly a challenge, I don’t see the connection between the statements here and anything to do with storage or processing resources. If the point is that packing significant computing resources into a near-detector farm is difficult, that should be said. Otherwise this challenge doesn’t seem to belong to this section.

L703 – Maybe “Additional R&D” instead of “addition RD”? Also, “so it will” rather than “to it will"

L755 – Presumably “DSLs”, not “DPLs”? But I don’t know to which this refers; either type of language might provide such primatives.

L786 – Given the ubiquity of mentions of “the grid” earlier in this document, it’s surprising that here “a batch system” is the most distributed system mentioned.

L836 – Why not a reference for this limitation?

L854 – Presumably “connection between QC and HEP”?

L859 – It seems very strange to talk about “HEP needs” at this stage. It is clear from this section that QC is more than a decade away from any use in HEP. Is there a clear reason why – 10 years out – HEP physicists should be contributing directly to QC software packages that exist today? Do we really imagine that the same packages will be used 10 years from now?

L878, and again at L893 – Only the cosmic frontier? Why?

L903 – Bad reference

L929 – “learning models” (plural)

L929 – Placing this assertion about ML here suggests that containerization will not work for machine learning. Why would that be the case?

Comment by: *Heidi Schellman,* *schellmh@oregonstate.edu*
Date: July 18
Report Version no.:CompFReport_preCSSversion
Section Number: see below
Line Numbers: *X Y Z*
Comment text

General comments:

1) Very little mention of the HEP software foundation. Should probably add some references where appropriate.
2) Please make this 12 pt type instead of 10pt, I had a lot of trouble reading this. We should be saving eyeballs, not paper at this point. Funding agencies would reject this without review for the 10pt type.

Page 3 4:09 AM Yesterday

Replace: *“pearturbative”* with *“perturbative”*

Replace *“particular machine”* with particular *“architecture”*

Line 59 add in for **theoretical predictions** and data analysis.

FIgure 0-1 do we mean “**only** meant to be illustrative”

Section 3.1 - should mention supernova calculations. Not certain which experts to ask but they are important for us.

Line 316 - add **safety** to the list of things you use machine calculations for (FLUKA/MARS)

Line 370 ;; **A wide range of other** experiments face similar computational challenges to process petabytes of data, including the corresponding simulation of background and signal processes **across a wide range of energy scales, from TeV muons to optical photons traversing meters of complex materials.**

(Side comment - IceCube and T2K are HUGE and DUNE is the box ATLAS came in)

Line 659 : run for many years **and require asynchronous bursts of processing for commissioning, fast calibration and parameter estimation.**

Line 754 what does “that” refer to? Jupyter? ROOT? Clarify.

Line 893 - Why just cosmic frontier? This seems to be a need across frontiers. Right now there seems to be no standard place for US experiments to put their data to make it accessible. I’d extend this one.

I’d be interested in helping with the training part if help is needed.

Comments by: Julian Borrill, jdborrill@lbl.gov
Date: 7/18/22

Do you agree with (or have any comments on) the main recommendations in the Executive Summary? These are the main messages that CompF plans to send to P5.

- **Continuous development**. Broad agreement, particularly wrt software that is not tied to a particular project, and even then only during its construction and operations phases. Two specific examples from the CMB community are (i) all of the software to go from single-frequency maps (which are a project deliverable) to science (which is a collaboration deliverable), and (ii) cross-correlation software, for example LSSxCMB, which effectively lies outside of either domain because it spans both.
- **Cross-cutting software/computing**. Broad agreement, with the caveat that ASCR funding is almost always focused on cross-cutting development, but typically looking for a wider span than is feasible for much of our work.
- **Heterogeneous computing**. Partial agreement, since the CMB community has a long history of evolving its software with HPC or HTC systems (depending on the experiment/data analysis team), and is now trying to align both. For CMB-S4, for example, supporting this evolution for the most computationally intensive codes is explicitly built into the WBS with significant resources earmarked. Also the second bullet is confusing since it seems to suggest that parallelization does not happen on traditional

CPU-based systems. I presume the intent here is to convey the challenge of coding for the novel architectures that are being fielded to squeeze out the last few FLOPS per Watt in the post Moore's Law world of energy-constrained computing.

- **Training & career paths**. Broad agreement, although there are notable and growing exceptions (eg. UC Berkeley's Computational & Data Science concentration). A particular challenge from a DOE perspective is that, unlike lab Physics Divisions and HEP, Computer/Computational Science Divisions get no base (research) funding from ASCR. In addition (and related) many lab career computational positions are for engineers rather than scientists, which changes the funding profile eg. for projects.

From a big project perspective, I would also like to see an Executive Summary comment on the existential need for (i) sufficient resources (cycles/storage/bandwidth) and (ii) multi-year allocations, maybe one for construction and a second for operations.

Do you have any comments on any of the numbered sections that are important for your experiment?

1. Introduction
    - Lines 43/50: Since we are discussing the next decade, we should emphasize that GPUs are only the current attempt to address the end of Moore's law, and by the end of the decade we may be looking at some entirely new and as-yet unimagined architecture (and learn the cautionary lesson on many-core, where huge amounts of work porting/optimizing codes was rendered redundant by a commercial vendor decision). We might also note that HPC systems evolve stepwise, but HTC systems continuously.
    - 106/107: rapid advances/rapidly growing.
2. Experimental Algorithm Parallelization
    - 155/156: CMB has been HPC focused for many decades, with hundreds of analysts from almost all experiments using a community-wide allocation at NERSC.
    - 179/182: Not just statistically noisy, but eventually systematics-dominated. An additional dimension here is that as we drive down statistical uncertainties with ever larger datasets we have to control systematics ever more precisely too, requiring more R&D on these large data volumes.
    - 196/198: I would say the the biggest challenge for CMB experiments is to generate and reduce a sufficient quantity of mock raw data sets of sufficient realism that we can (i) validate and verify the experiment design (including its data reduction and analysis), (ii) perform the necessary R&D on systematics mitigation algorithms, (ii) support Monte Carlo analyses. None of this will use field computation, and the simulations are necessarily more efficient on HPC than HTC systems.
    - 223/236: We also have to find a way to pay more than lip-service to the need to bring the skills of applied mathematicians, computer scientists, computational scientists, and domain scientists together in a sustainable way. I'm not sure if it

belongs here, but the big projects also need multi-year commitments across all their resources (cycles, storage, bandwidth) from the national facilities rather than the current annual allocations of cycles.

3. Theoretical calculations & Simulation
    - 249: “Automation of memory hierarchy” smells a bit like “the compiler will do it all” … nice idea, but really hard to imagine it being both general and efficient.
    - 261: SciDAC has been great for the supported domains, but has left a lot of the rest of us behind.
    - Generally I’d like to see some identification of the trade-off between generality and efficiency in both algorithms and implementations.
    - 3.1: Cosmic Calculations seems exclusively focused on LSS/Galaxy formation simulations. Where does the generation of mock datasets sit? Or things like MHD simulations to understand galactic dust and synchrotron statistics to enable their removal from CMB datasets?
4. Machine Learning
    - What about ML-derived data selection? observation strategy!?
5. Storage & Processing Resources Access
    - 610/611: While I understand the reason for this, it does leave a critical hole in the report. Even at a flagship HPC center, local data management between scratch, persistent spinning, and archival storage is still a major bottleneck - both from insufficient quotas and limited bandwidth. At very least we could point to the centers’ own analyses of storage requirements and their plans to address these, and support such efforts as being mission-critical.
6. End-User Analysis
7. Quantum Computing
8. Reinterpretation & Long-Term Preservation
    - The CMB community has been moving towards a model where the raw data and the software used to reduce them are released along with the reduced data products. However, while the latter can straightforwardly be served from data archives (eg. NASA’s LAMBDA), the former also require significant computational resources to make use of them. We are attempting to provide this via the community repo at NERSC, but that is then set against the resource needs of the experiments themselves.
9. Personnel & Training

Are there any computational topics that are important to your experiment that are missing in this report?

My only concern is the lack of emphasis on the sufficiency and long-term accessibility of computational resources across the board (cycles, storage, bandwidth). This is critical enough to our future that I would like to see it as an additional bullet in the Executive Summary, even if only as a broad statement.

Is there any other feedback that you want to provide to the CompF conveners?

Thank you!

Comment by: Philip Ilten, philten@cern.ch
Date: 18/07/2022
Report Version no.: v0

Section Number: 3.4
Line Numbers: 408
It would be good if a more inclusive LHC statement could be made here, something like "estimated to be 5 - 12%, with similar or even more substantial requirements expected for ALICE and LHCb."

Section Number: 3.4
Line Numbers: 421
This seems like a slightly odd example to choose here, particularly given the timeline for Snowmass. There has been substantial progress on negative weights, in the context of MC@NLO with aMC@NLO-Delta (https://arxiv.org/pdf/2002.12716.pdf), with a practical release of the code immanent. The example of event weights is still relevant, e.g. parton shower variations.

Section Number: 3.4
Line Numbers: 432
It is probably also worth mentioning ML in hadronization as well: 2203.12660 [hep-ph] and 2203.04983 [hep-ph].

Section Number: 3.4
Line Numbers: 437
There are other long-term projects like this that might be worth mentioning here, e.g. CTEQ, which are not equivalent but are also good models for training.

Section Number: 3.4
Line Numbers: 433 - 436
I think that it is also important to emphasize here that while there is funding support for new features, there is very little support for maintaining or improving established features via software development.

Thanks for putting together a great report!

- Phil

Comment by: *Heidi Schellman, schellmh@oregonstate.edu*
Date: 7/22/2022
Report Version no.: v0 - actually the recommendations shown on 7/22
Section Number: XYZ
Line Numbers: *X Y Z*
Comment text

Feedback on the recommendations; Comment on bridge vs. joint position. Fermilab calls them joint, apparently BNL calls them bridge. I would suggest saying "short-term joint positions to facilitate the hire of junior faculty" . FNAL has been very successful with these and sounds like BNL as well. As an associate Dean and dept. Head I negotiated several of these with FNAL. THey can actually end up being longer term but the normal term is until tenure. It also saves the labs a boatload of $ as the overheads are university level, not lab level.

I agree that QC needs to be mentioned by name in the findings or recommendations. But probably mainly to comment on the promise.

Liz Sexton has commented that calling out the existing joint projects HEP-CCE, SCIDAC, IRIS-HEP, HSF as examples of success that need to be built upon.

Heidi: I would comment that extending those endeavors to include the smaller scale experiments including DM and the short baseline neutrino experiments.

Comment by: Amy Roberts, amy.roberts@ucdenver.edu
Date: 7/22/2022
Comment on the recommendations shown on 7/22, recommendation on career paths

Another example of successful creation of positions for computing: the Space Telescope Science Institute (STScI Home). I spoke with Jinmi Yoon and she may be willing to talk with us more about this, or refer us to people who can talk with us about what the funding model looks like.

Amit Bashyal (abashyal@anl.gov)
Comment as a (sort of novice in this field….) on slide 9 of
https://indico.fnal.gov/event/22303/contributions/246924/attachments/157819/206668/CompF-Report-Session-V1.pdf

Wondering if. Position with split FTE (for early careers like postdocs) for physics as well as projects that builds expertise on HPC resources etc. Probably only possible for experiments like CMS/ATLAS/DUNE (?). Probably this will build person power that can develop software that will

rely on future technologies/HPC resources and also develop skeleton for physics tool which could become the basis for future analysis in many years too come.

Comment by: Amy Roberts, amy.roberts@ucdenver.edu
Date: 7/22/2022
Comment on the recommendations shown on 7/22, recommendation on heterogenous HPC resources

A comment on the need for traditional, CPU compute being needed in the future.
This is very true for my collaboration (CDMS). We have a significant chunk of code that runs only on CPU and I can't imagine we'd ever get funding to switch it over to a different method because (1) the code is specific to our collaboration and (2) we can get away with just running it on CPU without limiting our science.
por

Comment by: Axel Huebl, axelhuebl@lbl.gov
Date: 07/22/2022
Comment on executive summary and recommendation draft for "3. We are not prepared to make the most of heterogeneous computing resources."

This comment has been discussed today and is not a R&D task nor in immediate danger that CPUs vanish. (The sub-points also neglect any progress in the last 5-10 years from SciDAC and ECP, which we can take as lighthouse examples and reuse/expand modular software and algorithms from.)

Hardware evolves and we have a never-seen explosion of commercially available hardware designs, continuing to evolve in the coming years (ARM/RISC CPUs, x86 CPUs, GPUs, FPGAs, ASICs, neuromorphic computing chips, TPUs …). Our recommendation should not be to focus on the problem that some algorithms are hard to port to GPUs - ignore all the other options, stay with traditional CPUs. If we do that, we will slow down software innovation and miss opportunities in emerging computing hardware.
Our recommendation might better be to investigate innovative algorithms (e.g. time-parallel PDE solvers, revisit algorithmic approaches, apply AI/ML methods and differentiable algorithms, among others), redesign approaches to solve branchy and traditionally serial computing tasks and map them to new, heterogeneous hardware (e.g. maybe a GPU does not solve your problem, but an FPGA or ASIC might).

Comment by: Ken Herner, kherner@fnal.gov
Date: 7/22/2022
Report Version no.:
Section Number: -
Line Numbers: -

Regarding the recommendation to create the Coordinating Panel for Software and Computing (CPSC) board, it would be good to include some mention of how experiments can (and should) engage with the board from the beginning so we don't end up with a large disconnect down the road between what the board thinks experiments should do and what experiments think they should do. This document probably isn't the place to get too detailed with the board structure but I would at least include something about the vision for direct experiment participation.

Comment by: Alex Himmel ahimme@fnal.gov
Date: 7/22/2022

Please number, letter, name, or otherwise clearly identify the key recommendations in the CompF report. I would like to reference them from the Computing section of the NF report, and I think that this kind of cross-referencing will help the recommendations carry greater weight.

Comment by: Maria Elena Monzani, monzani@stanford.edu
Date: 7/22/2022
Comment on the recommendations shown on 7/22

At least 2 of the recommendations use jargon that is misleading:

- #1: "Continuous development" has a precise meaning in the software development cycle and can for example be confused with "Continuous deployment". I suggest rephrasing to "long-term support" or "ongoing support". Also, that recommendation is missing "experiment-independent" or "experiment-agnostic". That is an important qualification, because we don't get told to go back to the experiments (or operations funds) to ask for support for software maintenance. Also, "modernization" covers for example new architectures (GPUs, etc.), so we should totally keep it in there.
- #4: "Bridge position" is jargon from one specific example (which is a good example by the way) and could be misinterpreted as meaning "temporary support for CS specialists who are looking to get another job". We should rephrase it to something like "Joint positions", "Multidisciplinary positions", or "Interdisciplinary positions". Speaking of which, we should highlight that we are looking for people with dual CS/physics expertise. By the way, the recommendation as written misses out on the crucial role of multidisciplinary scientists at the national labs, so we may want to make it more inclusive (I have the feeling that the folks with dual Lab/university faculty appointments are doing lots of management and "stakeholder management", but no "real" software development work, so it doesn't really solve our problem of expertise).

Comments from Kevin Pedro pedrok@fnal.gov
(trying to summarize verbal comments during the CompF parallel session on 07/22)

- On Recommendation 4 "bridge positions": this comes from the CompF2 report via USQCD and related white papers (based on the BNL model, as remarked by others). The recommendation is very concrete: DOE, through the national labs, pays for the first ~5 years of a new tenure-track faculty position. This incentivizes the universities to open

such positions, and then the presence of tenured faculty working on HEP computing increases the perception of the academic merit of this work. This should be expressed as clearly and directly as possible in the recommendation and the report.

- On Recommendation 1 “targeted investment”: this is too vague a request. We should be more concrete about establishing institutes or recognized collaborations for long-term (indefinite) support of common software tools. We do not want “targeted funding” to be implemented as “here’s a postdoc” or something similar that won’t be sufficient to achieve the goal. This is a gap in the current funding model: large experiment operations mostly supports the experiments’ internal tools and usage of (but not development or maintenance of) common tools. There is some support of common tools, but not enough. It was also pointed out that small experiments have small or nonexistent operations budgets and can’t provide the same kind of internal support.
- On diversity & inclusion: we included a paragraph on this in the CompF2 executive summary; feedback welcome. This was unfortunately not a topic that was addressed extensively earlier in the Snowmass CompF efforts and discussions. I think we should include this as a focus of any panel/board that is created, to further investigate these issues and come up with more concrete recommendations aimed at CompF (ideally in just a couple years) and based on the overall community engagement work from Snowmass.
- A comment I did not have time to make during the session (but which was briefly mentioned by others): replacing classical/rule-based algorithms with ML algorithms can lead to a solution that is automatically portable and acceleratable on coprocessors. This overall idea falls in a gap between CompF1 and CompF3 (it is mentioned in https://arxiv.org/abs/2203.16255 and implied in some CompF2 report discussions for detector simulation and event generators). We should see if there is a way to provide more emphasis on this cross-cutting idea in the CompF report.

A few additional comments posted later (July 24):

- The phrase “continuous development” in Recommendation 1 should be revised; the word “development” may make funding agencies think of “R&D” rather than basic “software development”, and support for indefinite R&D is not what we are requesting.
    - Liz Sexton-Kennedy suggests the term “sustainability” should be included.
- The funding agencies may respond more positively to our recommendations if we tie them more closely to needs for the HEP mission and goals.
- The Snowmass survey https://arxiv.org/abs/2203.07328 shows in Fig. 17 that the majority of respondents think development and maintenance of open-source software is underfunded in our field. It would be good for the report to cite this as an indication of community support.

Comments from Jan Strube:

Regarding the text about small experiments (p. 15): A challenge in my experience has been to get accounts for everybody in a team with the same computing platform, such that running each

other's code and accessing data is straightforward. I see a particular challenge for future detector R&D collaborations that are not associated with a CERN or FNAL experiment. This may seem like a trivial issue, but getting accounts for everybody at the same institution has been a big frustration in my experience.
In the discussion on Saturday morning, OSG was mentioned as a possible resource for such collaborations. It would be helpful to add a recommendation that funding be ensured for such an organization that provides a common platform for small collaborations even with little research funding (CPU hours could be provided opportunistically, for example).

Template:
Comment by: *Heidi Schellman schellmh@oregonstate.edu*
Date: 7/23
Report Version no.: Recommendations
Section Number: XYZ
Line Numbers: *X Y Z*
In recommendation 1, make certain to add that the software packages are generally cross-experiment so the payoff for support is even greater but funding is harder.

Comment by: *David Lange, david.lange@princeton.edu*
Date: 7/23
Report Version no.: 0
Section Number: 1 and 2

Overalll: I find that the current drat is formulated in a way that prove to be difficult to convey the challenge of the compF frontier to non-experts. For example, it would benefit from a very high level (summary) view of challenges/uncertainties in software and computing over the next decade and, importantly, what are the needs for each in order to reach the science potential of our experiments/facilities. There is a lot of what reads like "blue sky" R&D and little to tie back that research to gaps in current infrastructure. Being Snowmass, it seems to me that connecting this R&D to the science in a concrete way is important (and not difficult in most cases) . Eg, help our colleagues distinguish "hype" from what's important to their science.

Comments on section 1/2

- Please add the document version into the PDF itself
- 3 : Surprising title - a) it doesn't give the context of HEP and b) does not include software

- 40 : A reference would be beneficial here. Eg, what is a computing limitation? Simply money to buy compute? What experiments are computing limited in their physics program? (Eg, mine is not) It might be easier to discuss this in terms of a costly investment on top of the detector cost that is driven by physics needs.
- 43 : We do a disservice by saying the computing infrastructure is not part of the experiment. I guess this really means the detector apparatus itself.
- 46-49 : The text about heterogeneous hardware seems out of place here. It is indeed a challenge for the field, but not the only challenge. (As the current recommendations also reflect). Why introduce it here without the other important challenges?
- 56 : I've not seen online used interchangeably with edge (google doesn't help me either). Perhaps this is jargon of some subfield and should be avoided?
- 57 : Not all online computing is "ultra-low" latency. Might be worth capturing the range of decision making in experiments?
- 69 : To be parallel, this first sentence should be in the previous paragraph
- Figure 0-1 : I had understood the hardware trigger was out of scope, so perhaps 40MHz is the wrong rate to consider for HL-LHC detectors. However, if it is, there will be far more than 40TB/sec generated. Was this from a reference? But indeed, the document could benefit from a short paragraph on its scope of what "software and computing" is
- 80; The third topical group? (To be parallel)
- 148: . . ? Typo?
- 150: "package" seems like the wrong word here.
- 151: perhaps be less lhc specific in the examples?
- 153: Would be really useful to capture what does "substantial" mean (perhaps via reference)
- 155: This paragraph is very hard to follow. I think it needs to be broken up and given some context (perhaps by addressing my comment about 46-49). Eg, why do we want to use HPCs? (They are large DOE investments that are free to HEP to use); why do we want to use AI/ML (obvious, but motivate it), Define what a portability library is (jargon)
- 166: This sentence can use some semicolons to break up the long list
- 170: I have no idea what this sentence means "run as part of".... "Executed within them"?
- 176: Some success stories would help considerably to understand what this sentence refers to
- 181: This sentence appears to consider apples vs oranges in its use of "signal" between the two fields compared. I would suggest reworking it. Collider detector experiments certainly have signals buried deeply in noisy data.
- 187: Perhaps "has the potential to play" is more correct?

- 188: A reference would be useful (this is not at all the aim of using GPUs in the CMS trigger for example - but CMS has the goal of common online/offline algorithms from the start)
- 193: Static "science" seems like the wrong word. Objects?
- 198: "require" or be most cost efficiently done with?
- 204: The start seems general to neutrino experiments - but the rest is not.. maybe unify the approach to the paragraph
- 223: Much of the paragraph includes concepts not discussed previously. Seems like nothing should appear for the first time here.
- 233: Seems this "projects" concept mixes "projects" in experiments (or perhaps even CMS specific jargon?) with the concept of funded projects. This needs reworking to clarify what concept is meant in the US context.
- 235: I think the jargon to use here is sustainability. Sustainability is indeed partly the job of the funding agencies, but not only. HEP researchers should be using sustainable software approaches as they develop and deploy it. I find it hard to put all of the responsibility onto the funding agencies

Comment by: Maria Elena Monzani, monzani@stanford.edu
Date: 7/23/2022
Comment on the recommendations shown on 7/23 (small experiments session)

- Interplay between the “main” recommendation and the 4 topical recommendations: the way it is written is confusing, because it feels like the 4 topical recommendations flow from the first one. This is misleading, because the first recommendation goes to DPF, while the other 4 go to the agencies. We should separate the two sets more clearly.
- Lots of discussion on open software and open data. Open software makes it easier to “recycle” tools (crucial to the small expt community), open data helps making our results more robust. The Compf7 report (and its summary here), discuss *how* to do sw/data releases, but not policy. We should make a clear statement for or against, and highlight where there is and isn’t consensus.

Comment by: Axel Huebl, Jean-Luc Vay, Ji Qiang (LBNL) axelhuebl@lbl.gov
Date: 07/23/2022 - also sent by email on the same date
Comment on CompF v0 Section 3.2: Particle Accelerator Modeling

We have collected a few suggestions for the section "3.2 Particle Accelerator Modeling" that we would like to share with you for your consideration.

Most of the updates are parallel updates we do in the corresponding CompF2 report section, more precise wording in some sentences and a final sentence on a few recommendations that we described too briefly when translating from the text in CompF2 & whitepapers to the CompF v0 section.

Please see the following google doc link with highlighted changes (a .docx export with the same changes was sent by email):
https://docs.google.com/document/d/1YYyPrKFy2xdDsHtUbVtlpspxiC-V3m5wBa2d8LZf7Vc/edit?usp=sharing (please log in to google to see the colorful changes)

All the best,
Axel, Jean-Luc and Ji

Comment by Tulika Bose (tulika@hep.wisc.edu)
Date 7/23/2022
Refers to summary slides:

Overall, they look good and I think it's very useful to see the recommendations attached to the challenges - thank you for adding them!

A few comments:

Slide 6: the sentence "...working with...." does not include experiments. Is that intentional ?

Slide 7: who is going to provide the "targeted investment" here ? Are you asking the funding agencies to provide that ? I would also say that some level of support for these tasks is indeed provided by NSF/DOE at least for ATLAS+CMS operations programs but not at the level that is needed resulting in many items not being covered.

Actually, it depends on what you mean by "essential software packages". Things like an experiment's software framework is experiment specific and they should get funding for it. But the issue is more complicated for common projects such as ROOT, GEANT4. Looks like this slide covers both ? The challenge associated with their support can be different needing different solutions and the line about GEANT4 is probably more suitable for the following slide (#8)

Slide 8: USQCD is a good example. does HSF actually get funding as an organization ? If yes, then it;s a good example. If not, then it's confusing since the recommendation is about funding...

Slide 10: not clear to me what "bridge position" means - maybe you want to clarify that.

I think this slide should also talk about increasing diversity and broadening participation. This is a big challenge too in my opinion, especially for software & computing.

Comment by Ji Qiang (jqiang@lbl.gov):
Date 7/24/2022 about Recommendation 3: Support the coexistence of the CPU-based hardware and the heterogeneous resources with computing accelerators. The goal of computing is for science. It is critical to keep the less scalable scientific production software running on state-of-the-art CPU computers.

Template:
Comment by: *Christan Bauer*
Date: 7/24/2022
Report Version no.: V0

Executive Summary:
I find it very strange that an actual executive summary of the computing report is missing (what is called "Executive Summary" now is more a list of requests). In my mind, such a section is absolutely vital to make sure the Computing Frontier is properly represented in the final document. I understand the desire to point out shortfalls in the current program, but giving up the opportunity to summarize the successes of the past decade and the promise of the next decade seems like a huge missed opportunity to me. It seems to me that essentially all other frontiers have chosen to provide an actual executive summary. I propose to add an executive summary which summarizes each of the sections of the report in 2-3 sentences.

Quantum Computing:
I find the executive summary provided by the CompF6 conveners a much better worded summary of the possible impact of Quantum Computing over the next decade and beyond. It seems that the section in the main report is its own independent section, which talks about things that are not mentioned in the topical frontier report, while other things that are highlighted in the topical frontier report have not made it to this section.

Template:
Comment by: *Heidi Schellman*
Date: 7/26/2022
Report Version no.: V0

On page 3, the summary of critical areas.

xamples include:
– S&C for theoretical calculations, including perturbative QCD, and lattice gauge theory, accelerator physics, and cosmological simulations.
– Cross-cutting machine learning methodology and computing model development.
– Tools that transcend experimental boundaries (algorithms, common packages, etc.)

Needs to expand to include a 4th:
– common tools that form the international infrastructure for HEP data and processing: xroot, dcache, Rucio, grid tools, authentication

---

Comment by: *Seth R Johnson, johnsonsr@ornl.gov*
Date: 27 July 2022
Report Version no.: (feedback from the summary slides)

Having attended the CompF presentations and discussions in person last week, I was surprised at the synthesized recommendation on HEP computing architectures that Daniel presented on Monday. Specifically, the emphasized phrase "traditional CPU-based hardware should coexist with heterogeneous resources" seems to geld completely the already rather timid directive "to use heterogeneous resources effectively."

It is imperative to emphasize that, although the breadth and algorithmic complexity of HEP computing gives it unique challenges, we are already playing catchup to other compute-heavy sciences. More importantly, today's HPC clusters foreshadow the commodity architectures of ten years' time. Already we're seeing a shift in the industry toward low-powered RISC chips (such as ARM) with multiple coprocessors such as GPUs, NPUs, etc. Asking for investment in "traditional CPU-based hardware" now is like asking for "traditional single-core CPUs" back in 2010: just as traditional CPUs hit a plateau in single-core performance then, they have hit their limit for multicore performance, and the industry is moving on.

As efforts such as the Exascale Computing Project have demonstrated, adapting scientific codes to heterogeneous architecture is naturally much more difficult than upgrading a single-threaded code such as Geant4 to work with multiple threads. Nevertheless, HEP computing needs a concerted effort to *(i)* characterize the total computational requirements of its multitude of workflows and their constituent components, *(ii)* evaluate the difficulty and potential benefit of adapting each component to use GPUs and the like, and *(iii)* prioritize software development based on that impact. Without such an effort, we will certainly be leaving unused the enormous computational resources that will power the current and next generation of machines.

Comment by: Andrea Valassi (andrea.valassi@cern.ch)
Date: 29 July 2022
Report Version no: v0 (CompFReport_preCSSversion.pdf)

General comment: thank you for putting together this nice and comprehensive report!

Section Number: Title page
Line Numbers: 13
Typo: "Weurthwein" -> "Wuerthwein".

Section Number: Executive Summary page 3
Line Numbers: 19-20, paragraph 1
Maybe add "reengineering" between "modernization" and "maintenance"? It is important to point out that the effort needed is often very substantial and involves a profound redesign of the code (for instance, improving performance by implementing parallelism through multithreading or vectorization implies profound changes to memory structures and to looping algorithms). I think that "reengineering" suggests an active large effort while "modernization" and especially "maintenance" may be looked down on as relatively simple tasks.

Section Number: Executive Summary page 3
Line Numbers: 19-20, paragraphs 1 and 2
Examples in paragraph 1 start lowercase, in paragraph 2 uppecase. Make these consistent.

Section Number: Executive Summary page 3
Line Numbers: 19-20, paragraph 2
Typo: "theoreticaly" -> "theoretical".

Section Number: Executive Summary page 3
Line Numbers: 19-20, paragraph 3
I find the first bullet a bit weak "Most of our software runs on a particular machine making it difficult to use hardware accelerators and diverse computing resources like cloud, HPC, etc.". I would turn this around and make it somewhat stronger, for instance <<Aggressive R&D is needed to ensure that we can, as much as possible, efficiently exploit diverse computing resources like cloud and HPC, including multiple CPU architectures and hardware accelerators>>.

Section Number: Executive Summary page 3
Line Numbers: 19-20, conclusion, after page 3
"worldwide partners", you may add "like the HEP Software Foundation".
More generally, HSF is mentioned only twice in the references. I would strongly suggest mentioning it also in the main text.

Section Number: 1 Introduction
Line Numbers: 36
"computing is ubiquitous" -> "software and computing are ubiquitous" (do mention software upfront)

Section Number: 1 Introduction
Line Numbers: 50?
It would be useful to mention somewhere (for instance, in a short new paragraph before the last sentence in line 50?) that HEP is not alone in this problem, and also that the expertise required to address software and computing issues is not specific to HEP and is typically not even part of standard HEP training. This is mentioned in the conclusions of the European Strategy, but it's worth pointing this out here explicitly. One could add for instance <<One further difference with other HEP frontiers is that many of the challenges relevant to the Computational Frontier are also shared with other sciences and, in parallel, that addressing software and computing issues is not the main focus of HEP academic training. The collaboration and expertise of software and computing experts from other academic institutions or industrial partners may be essential to address some of the HEP specific issues in this area. Relying on and contributing to non-HEP community solutions such as open source software, in particular, may significantly boost the productivity of particle physics experiments while also increasing our immediate return to society. Training HEP physicists for modern software development is also a challenge.>>

Section Number: 2 Experimental Algorithm Parallelization

Line Numbers: 122
I would use uppercase for High Performance Computing centers.

Section Number: 2 Experimental Algorithm Parallelization
Line Numbers: 156
The comment on HTC vs HPC is true, but not specific only to section 2. There are some useful comments in lines 787-794 about HEP not needing tightly coupled supercomputers, maybe these can be mentioned higher up in a common part?

Section Number: 2 Experimental Algorithm Parallelization
Line Numbers: ~172
Maybe mention CPU vectorization somewhere? (data parallelism, not just multi/many core)

Section Number: 2 Experimental Algorithm Parallelization
Line Numbers: 203
What are "I/ML machine techniques"? Maybe "AI/ML techniques"?

Section Number: 2 Experimental Algorithm Parallelization
Line Numbers: 228
Typo: "roll" -> "role"

Section Number: 2 Experimental Algorithm Parallelization
Line Numbers: 235
See my previous comment about reengineering. Maybe add after "and maintenance phases" something like <<, >>?

Section Number: 3 Theoretical Calculations and Simulation
Line Numbers: 241?
One general comment here: before jumping to commonalities between the six domains, it may be useful to first mention that there are some large differences between the computational challenges of the six domains. For instance, I have the impression that accelerator simulations (maybe also cosmic calculations? CFT? lattice QCD?) may require a highly coordinated interplay of several tasks (as you typically obtain on an HPC with fast inter-node connections), while it is clear that things like matrix element calculations for different phase space points in event generators are completely independent of one another and can be done on one computing node of an HPC at a time. I note again that there are some useful comments in lines 787-794 about HEP not needing tightly coupled supercomputers, maybe here for theoretical calculations it would be useful to explain which tasks need which model ("HTC" or "HPC")?

Section Number: 3 Theoretical Calculations and Simulation
Line Numbers: 244
This sentence is surprising, because it is the first sentence in the overview of section 3 and because it does not seem sufficiently motivated by the six subsections: "On the hardware side, there is a need for faster HPC resources, ten times or more faster than planned Exascale computing systems". Expanding on my previous comment, there are certainly many examples of theoretical calculations (such as matrix element calculations in MC event generators) which use parallel "HTC" jobs and do not need faster HPCs. Again, it would be useful to clarify which theoretical calculations do need tightly-interconnected HPCs.

Section Number: 3.3 Detector Simulation
Line Numbers: 379-381
I am not convinced that this sentence is the best way to describe the situation: "Because detector simulation is naturally an HTC problem, nontrivial effort is required to adapt the necessary computations to the HPC environment". I think that "HPC" is a rather generic term that has different meanings for different people. When opposed to "HTC", the emphasis is mainly on HPCs being tightly interconnected systems: however, on an HPC center you always have the option of ignoring many-node setups, and work on one computing node at a time, in an "HTC" fashion, and this is not a problem for detector simulation. The more relevant problem in my opinion is that the most recent HPCs take most of their computing power on GPUs, and it is porting to GPUs that is difficult for detector simulation, especially because there is a lot of branching involved, often of stochastic nature. I would rather say <<Because detector simulation involves a lot of software branching, often of stochastic nature, nontrivial effort is required to adapt the necessary computations to GPUs>>. Another, somewhat less complex issue in porting to "HPCs" or supercomputers is that these often use recent and somewhat exotic CPU architectures (ARM, Power9 etc), which sometimes are not supported by the existing software, but this is not a major problem, or certainly it is not specific to detector simulation.

Section Number: 3.3 Detector Simulation
Line Numbers: 387
When discussing GANs and other generative ML techniques, it may be useful to note that the validation of physics results (i.e. making sure that the physics quality is enough) is one of the challenges.

Section Number: 3.3 Event generators
Line Numbers: 426-427

This is certainly true "Historically, there has not been a single obvious locus to coordinate common activities". I would however mention MCNet (already mentioned lower down) and the HSF generator WG in this context. You could add for instance <<Historically, there has not been a single obvious locus to coordinate common activities, even if MCNet and more recently the HSF Generator WG have provided very beneficial fora for stimulating common activities, especially around training and computational efficiency, respectively."

Section Number: 3.4 Event generators
Line Numbers: 428-429
I would modify this sentence "Projects to reduce the computational burden of event generation, for example by adapting to use GPUs, need a substantial increase in effort in order to be successful" to add two points: first, that matrix element calculations are a perfect fit for CPU vectorization and GPUs; second, that there are already efforts in this area that are producing promising results. I would write for instance: <<For complex LHC physics processes, matrix element (ME) calculations take up more than 90% of the computing time spent on event generators. For this reason, speeding up ME calculations is the priority in several ongoing R&D projects. Fortunately, computing MEs is a task that leads itself naturally to exploiting data parallelism through CPU vectorization (SIMD) and on GPUs, because the same calculation can be executed in parallel and in lockstep for many phase space points at a time. Very promising results have already been obtained in the ongoing port of Madgraph5_aMC@NLO to GPUs and vector CPUs[a,b,c]. A substantial increase in effort to port this and other generators to GPUs is nevertheless needed.>>. The references I suggest to add are the following (if you need to choose only one, take b which is a published paper):
[a] HSFWS2020: https://indico.cern.ch/event/941278/contributions/4101793/
[b] vCHEP2021: https://doi.org/10.1051/epjconf/202125103045
[c] ICHEP2022: https://agenda.infn.it/event/28874/contributions/169193/

Section Number: 3.5 Continuum Field Theory Calculations
Line Numbers: 440 (title, Continuum Field Theory) and 467 (Conformal Field Theory)
As a complete non-expert in these issues, I am confused by many points. First and foremost, the title describes "Continuum Field Theory", but the word "continuum" appears nowhere in the text of section 3.5: instead, lines 467 mention "Conformal Field Theory". Should the title be "Conformal Field Theory Calculations"? I am also a bit confused why the text in lines 441-448 is here and not in section 3.4, since many of these issues for NNLO/N3LO are often described for event generators. Maybe you should add one sentence to describe what this section 3.5 is all about (e.g. numerical integrations and semi-analytical methods?).

General comment (triggered by section 3.5)
It would be useful if all the text was also easily readable by non experts.

Section Number: 3.6 Lattice QCD
Line Numbers: 495
The reference is wrong (there is a question mark instead of a number).

Section Number: 3.6 Lattice QCD
Line Numbers: 500-501
"The massive vector parallelism of lattice gauge theory is, in principle, amenable to GPU and possibly other acceleration". Is it also a good fit for (and/or does it already exploit efficiently) CPU vectorization (SIMD)?

Section Number: 4 ML
Line Numbers: 521-522
"While industry is driving many of the developments in machine learning, HEP has unique challenges that require dedicated solutions." I completely agree with that. I would suggest mentioning that dedicated research on the training and evaluation metrics used for HEP is needed, as we should not be taking as-is tools that have been developed to solve very different problems.

Section Number: 6 Analysis
Line Numbers: 734-735
"There are two major ecosystems in relevant to data analysis in HEP". First, this sentence makes no sense, please remove "in". Second, I suggest adding immediately "There are two major ecosystems relevant to data analysis in HEP, based on ROOT and on Python respectively". Otherwise the following sentences are a bit confusing (the Python ecosystem is only mentioned much later).

Section Number: 6 Analysis
Line Numbers: 787-794
As mentioned above, there are some useful comments in lines 787-794 about HEP not needing tightly coupled supercomputers, maybe these can be mentioned higher up in a common part, not just for analysis? Then anything specific to analysis should be explained here.

Section Number: 8 Quantum Computing
Line Numbers: 833-834
"A number of benchmark problems have emerged which have focused the QC for HEP community and which are particularly promising. These include lattice gauge theory,

event generation, and data analysis." Can you add some references to back these statements?

Section Number: 8 Quantum Computing
Line Numbers: 854
"the connection between QCD and HEP is not one way"... you mean QC, not QCD here?

Comment by: *Peter Boyle, pboyle@bnl.gov*
Date: 11 Aug
Report Version no.: v0

Section Number: 3
Line Numbers: 258-260
Comment text

> *Lab-supported software efforts including development, maintenance, and support for common software tools in the areas of accelerator modeling, detector simulation, physics generators, lattice and continuum field theoretical calculation and cosmic simulation must be strengthened*

Section Number: 3
Line Numbers: 245
Comment text

Was asked by Ben Nachman to drop the contents of an email into these comments. The definition of “right-sized” CPU provision in the CompF2 report was: **“as large as required, as small as possible”**. This qualification is important to propagate and might alleviate some of the concerns raised in the CompF discussion.

Email comment:

> *The parenthesis in CompF2 recommendation about “as small as possible” might allay concerns?*
>
> - *Support of right-sized CPU clusters: There should be a right-sized **(as large as required, as small as possible)** provisioning of general purpose computational cores with high performance memory for important algorithms that do not easily map to acceleration.*

*I'd also point out there are ****significant*** moves on CPU front with similar novel memory directions.*

CPU only

- *Intel SapphireRapids with HBM,*
- *AMD with 'vertical cache' true 3D stacked server chips*
- *Fujitsu's older Fugaku/ARM with HBM*

Hybrid/converged devices "accelerated processor units"

- *Nvidia "Grace" ARM multicore looks interesting both as a server CPU also as a CPU-GPU integrated option*
- *AMD MI-300 is an integrated CPU-GPU multichip package.*

Non-GPU exotic devices

- *+ non-GPU spatial acceleration / FPGA / dataflow possibilities*

I think there is a risk of overcommitting to an 'all GPU' story in a Snowmass long range discussion and describing the current ASCR computers as
'obviously' the future mix. There will be GPU's for the largest part of the computing, but I think signing up to a line that this will be to the exclusion
of all others is actually damaging to science, productivity and prevents key things.

Our high order perturbative friends don't even use floating point in many things they do.

I put a little effort into surveying the trends in my TF / CompF Snowmass slides (I was given the title "Computational Trends in LGT")

https://indico.fnal.gov/event/22303/sessions/20645/#20220721