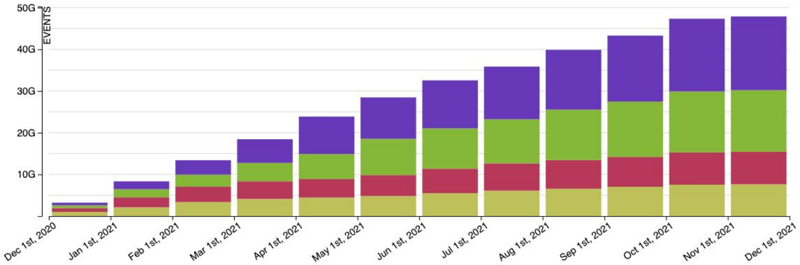
LHC evgen production experience
with input from ATLAS & CMS experts

Andy Buckley
University of Glasgow

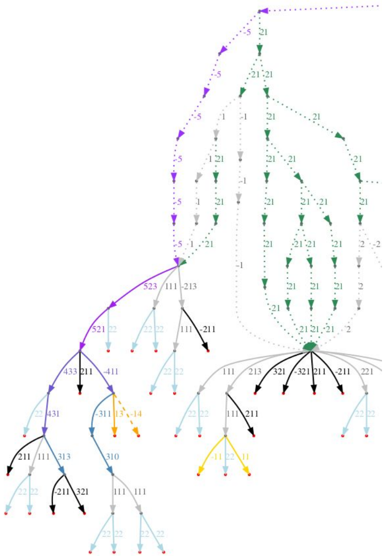
ECFA Higgs Factories, MC Generators Meeting
10 November 2021

LHC experience of MC production

❖ **Both ATLAS and CMS batch MC event generation in *campaigns* of O(50 Gevt)**
  ➢ not every year: often reuse+extend evgen



❖ **Running generators at scale introduces exciting new failure modes!**
  ➢ rare numerical issues
  ➢ configuration mistakes *very* costly
  ➢ requires serious software and configuration management

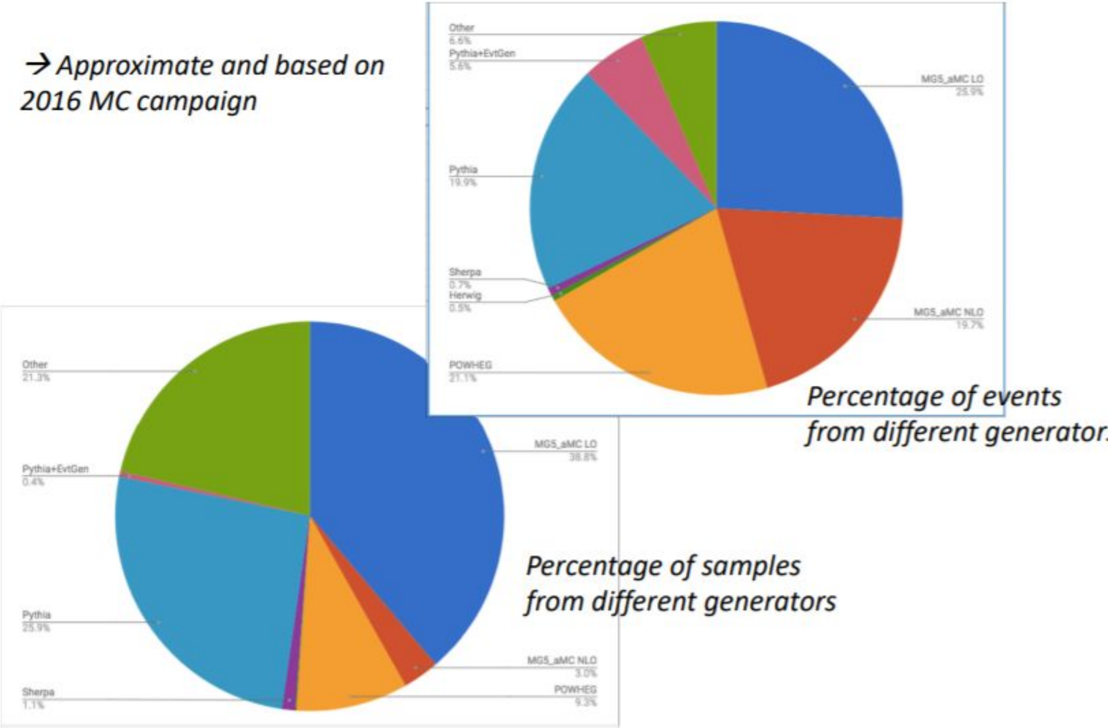


❖ **MC generation is a part of LHC data processing where natural CPU scaling is particularly strongly *upward***
  ➢ analysers have accustomed to MC being a good proxy for data. *Always* demand for next order in precision
  ➢ next order in precision typically costs an *order of magnitude* more CPU!
  ➢ MC evgen has not been “cheap” for some time
  ➢ sometimes the right answer is “no”... match precision to requirement

Generator balance

- **CMS has a particularly strong reliance on MG5_aMC and Pythia**
  - 97% Pythia showering!
  - few percent Herwig and Sherpa variations

- **ATLAS a much broader set of generators:**
  - Sherpa NLO for *V*+jets, and *VV* processes
  - Powheg+Pythia for *tt*
  - MG5 LO for most BSM



from Efe Yazgan, CMS

Code management and integration

❖ Code and build management

- ➢ LHC experiment software mostly built on LCG software bundles for base architecture and libs
- ➢ Generators included via GenSer project: take release tarballs, (patch) & perform basic tests.
  Uses CMake-driven system to build for multiple architectures
- ➢ Used by ATLAS and LHCb, tarballs by CMS, none by ALICE
- ➢ Experiment frameworks need "glue" packages to pick up compiled generators.
  Library version compatibility, e.g. FastJet & HepMC, not always coherent…
  ⇒ ATLAS former coordinators split on whether better to rm the intermediate… CI, containers?

❖ Experiment-framework interfacing

- ➢ Dedicated packages for each generator — configurations not all programmatically friendly (Powheg, MG5); matching, re-hadronisation, etc. hooks for many different Pythia modes
- ➢ Documentation: prefer repo READMEs to wikis!!
- ➢ Algorithm chains designed to allow use of afterburner postprocessing: now mainly for precision samples only — built-in QED, tau decays, etc. pretty good. ATLAS: EvtGen standard, not unproblematic
- ➢ Post-proc event-graph testing and fixing: look for common problems like unknown PIDs, broken graphs, unexpected displaced vertices, E & p imbalances. Exit above (typical, configurable) failure rate ~1%
- ➢ Commissioning of new generators using regression testing against physics-analysis suites (Rivet-based JEM in ATLAS)

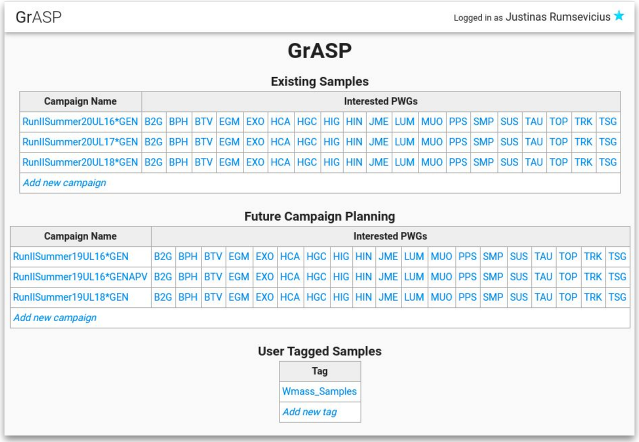
Sample-configuration management

❖ **Python job options in both ATLAS and CMS**

- ➢ O(10,000-1,000,000) sample configurations: jet slices, heavy-flavour filtering, BSM grid-scans, other enhancement-biasing
- ➢ New ones *will* mostly be written via copy & paste by non-experts: validation process
- ➢ Vigilance needed to identify common elements and manage common JO snippets. Chain snippets for e.g. standard EW params, tunes, modes, …

❖ **Managing sample requests & production status**

- ➢ Spreadsheets, JIRA, … "keep it simple. I'm sure a GitLab issue would work just fine"
- ➢ CMS more sophisticated than ATLAS: dedicated Web apps, e.g. GrASP, vs Twiki+GSheets+JIRA+… ⇒
- ➢ Use connected systems for production management and sample db — ATLAS doesn't, and it's awkward



GrASP Logged in as Justinas Rumsevicius

GrASP

Existing Samples

Campaign Name	Interested PWGs																		
RunIISummer20UL16*GEN	B2G	BPH	BTV	EGM	EXO	HCA	HGC	HIG	HIN	JME	LUM	MUO	PPS	SMP	SUS	TAU	TOP	TRK	TSG
RunIISummer20UL17*GEN	B2G	BPH	BTV	EGM	EXO	HCA	HGC	HIG	HIN	JME	LUM	MUO	PPS	SMP	SUS	TAU	TOP	TRK	TSG
RunIISummer20UL18*GEN	B2G	BPH	BTV	EGM	EXO	HCA	HGC	HIG	HIN	JME	LUM	MUO	PPS	SMP	SUS	TAU	TOP	TRK	TSG
Add new campaign																			

Future Campaign Planning

Campaign Name	Interested PWGs																		
RunIISummer19UL16*GEN	B2G	BPH	BTV	EGM	EXO	HCA	HGC	HIG	HIN	JME	LUM	MUO	PPS	SMP	SUS	TAU	TOP	TRK	TSG
RunIISummer19UL16*GENAPV	B2G	BPH	BTV	EGM	EXO	HCA	HGC	HIG	HIN	JME	LUM	MUO	PPS	SMP	SUS	TAU	TOP	TRK	TSG
RunIISummer19UL18*GEN	B2G	BPH	BTV	EGM	EXO	HCA	HGC	HIG	HIN	JME	LUM	MUO	PPS	SMP	SUS	TAU	TOP	TRK	TSG
Add new campaign																			

User Tagged Samples

Tag
Wmass_Samples
Add new tag

❖ **Distribution**

- ➢ JO updates far too frequent to include in sw releases. Sync via CVMFS or tarballs. Need versioning: configs need to be exactly repeatable, for sample extensions

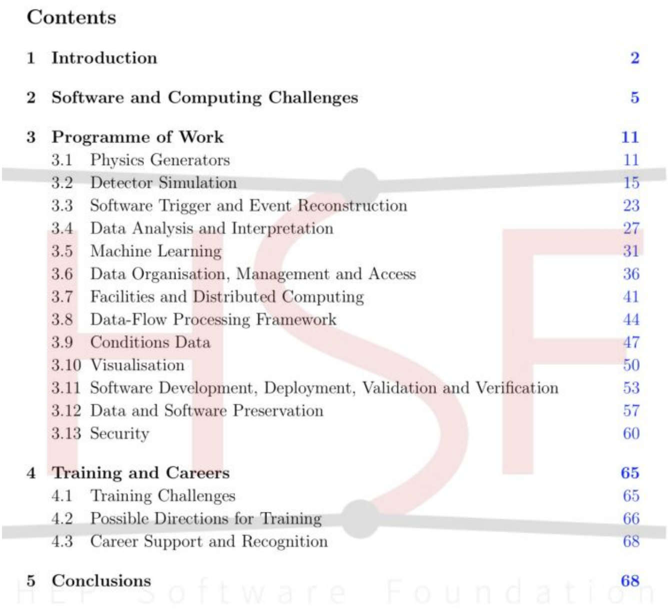
Generation practicalities

- **Logistics of bulk NLO production**
  - With "generator generators", job splitting to O(1kevt) ⟶ don't waste time rebuilding the model and remapping the phase-space (integration) in each job
  - Prebuild "gridpacks" storing results of integration: produced privately on HPC. Gridpack distribution with JOs / via CVMFS
  - LHE generation and tracking correspondence to showered HepMC event (esp. with post-shower event-filtering). Embed in HepMC3?
- **Weighting, filtering and enhancement**
  - Systematics weights: currently O(100) for ME scales, PDFs, sometimes shower vars
  - Post-hoc filtering: focus samples on flavour combinations & phase-space of interest
  - Increasingly "enhancing" phase-space coverage with biased sampling and counter-weights: adaptive samplers in multileg codes "learn" biases, so efficient
- **Data formats**
  - Automatic persistency is overrated — if any risk of change, use dedicated TP converters to persist key event info and handle compatibility. Annoying, but…
  - Downstream analysis formats: reduce event graph to collections of standard e.g. (many different) truth jets with truth flavour-tags dressed leptons, truth MET, etc.

Physics content & communication

- **Experiment generator experts rely on MC author input**
  - Get direct bug/task reporting with authors
  - Need regular interaction — specialist teams (and meetings)
    - Note incentive mismatch, cf. HSF white papers: generator-author motivations and rewards are from "theory world" — helping experiments is not #1!
  - Need ability to supply standalone configs to authors: they can't do anything with JOs ⇒ design interfaces with ability to dump standalone steering files
  - don't try to be too clever!
    - cf. BSM-grid magic dataset IDs and ~empty JOs 😬

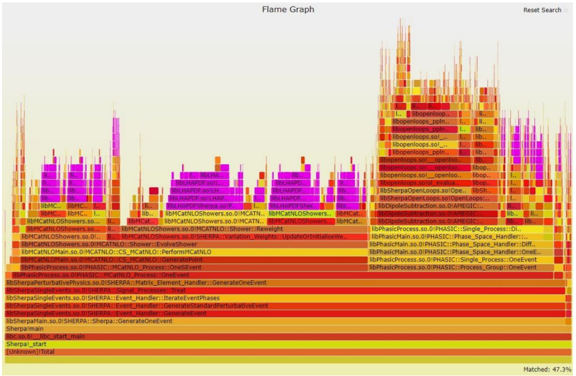
- **Standardising**
  - Work with authors to standardise: formats, systematics weight structure, PDG codes, … evolving and enforcing standards makes everything better/clearer in the long run


Contents

Computational performance

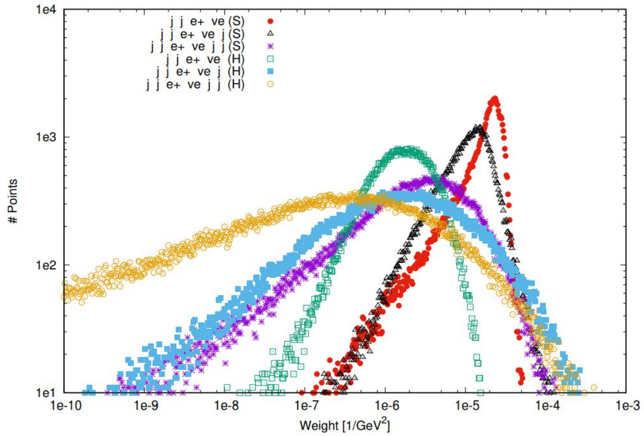
Bulk of CPU comes from multileg NLO *V*+jets, *tt*

- Don't forget associated packages, e.g. LHAPDF
- 40% of Sherpa CPU in PDFs! Joint exp/ph projects to address:
  - reduce PDF calls
  - speed PDF computation
  - optimise transcendental functions, specialise, …



First-world problems

- Unweighting (mismatch of proposal φ^N to real ME(φ^N)) the biggest problem. Worse at NLO. ML-based proposal densities?
- Negative weights: double-dilution of stat power. Use newer matching schemes, positive-weight resampling…



New architectures (?)

- GPUs, other accelerators, vectorisation: nascent efforts, not mainstream, focus on ME since shower etc. trivially parallelisable
- Focus on new gen strategies using HPC (particularly US)
- Opinion: unclear how relevant some of these options are: some focus on GPUs is because there's money/kudos there. Logistics?

Summary

Generation at scale requires great care

- Majority of core background processes are needed at high precision:
  not cheap, and trend is toward more expense
- Thousands of configurations to manage
- Mistakes are costly!
- Investment in people and functional, well-integrated systems

Communication with authors essential

- Physics content is very complex, many hidden wrinkles
- Design interfaces to enable communication with authors.
- Incentivise rapid responses, provide dev person-power to help

Performance still an issue

- MC authors are not strongly *incentivised* to solve expt problems!
- Some important performance developments… but the problem will keep coming
- Plan for hands-on performance and API work