GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

AthSimulation on Aarch64
(and potentially other architectures)

Joshua Wyatt Smith

II. Physikalisches Institut

Supervisor: Arnulf Quadt

Technical Interchange Meeting Glasgow
07 June 2016

BMBF-Forschungsschwerpunkt ATLAS-EXPERIMENT
Physik bei höchsten Energien mit dem ATLAS-Experiment am LHC
FSP 103 ATLAS
INSPIRE
Bundesministerium für Bildung und Forschung

Contents

Introduction

- Lots of ATLAS workloads not necessarily CPU intensive (apart from MC Generation)
  - Don’t need high Gflops as top requirement
- How do specifically designed power efficient architectures perform?
- Building for one architecture can be unhealthy
  - Compiler specific code emerges
- More architectures are always being developed
- As a result ATLAS explores different hardware (i.e. ARM) and software (i.e. clang/llvm)

Overview from last time

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

ARM: One such power efficient architecture

- ARM = Advanced RISC Machine
  - RISC = Reduced Instruction Set Computer
- Found in ~95% of smartphones and tablets
- 32- and 64-bit (Aarch64) available
- A company that sells its rights to its Intellectual Property.

Currently using

HP ProLiant m400 Server Cartridge

- AppliedMicro™ X-Gene™ 2.4GHz, ARMv8 64-bit cores (8)
- 64 GiB DDR3
- 32 KiB L1/core, 256 KiB L2/core pair, 8MiB L3
- Ubuntu 14.04 and CentOS 7

New Evaluation Prototypes!

- 2.1GHz, ARMv8 64-bit, A57 cores (32)
- 128 GiB DDR3
- Supposedly pretty impressive… we will see :)
- Ubuntu 14.04 and CentOS 7

?

Overview from last time

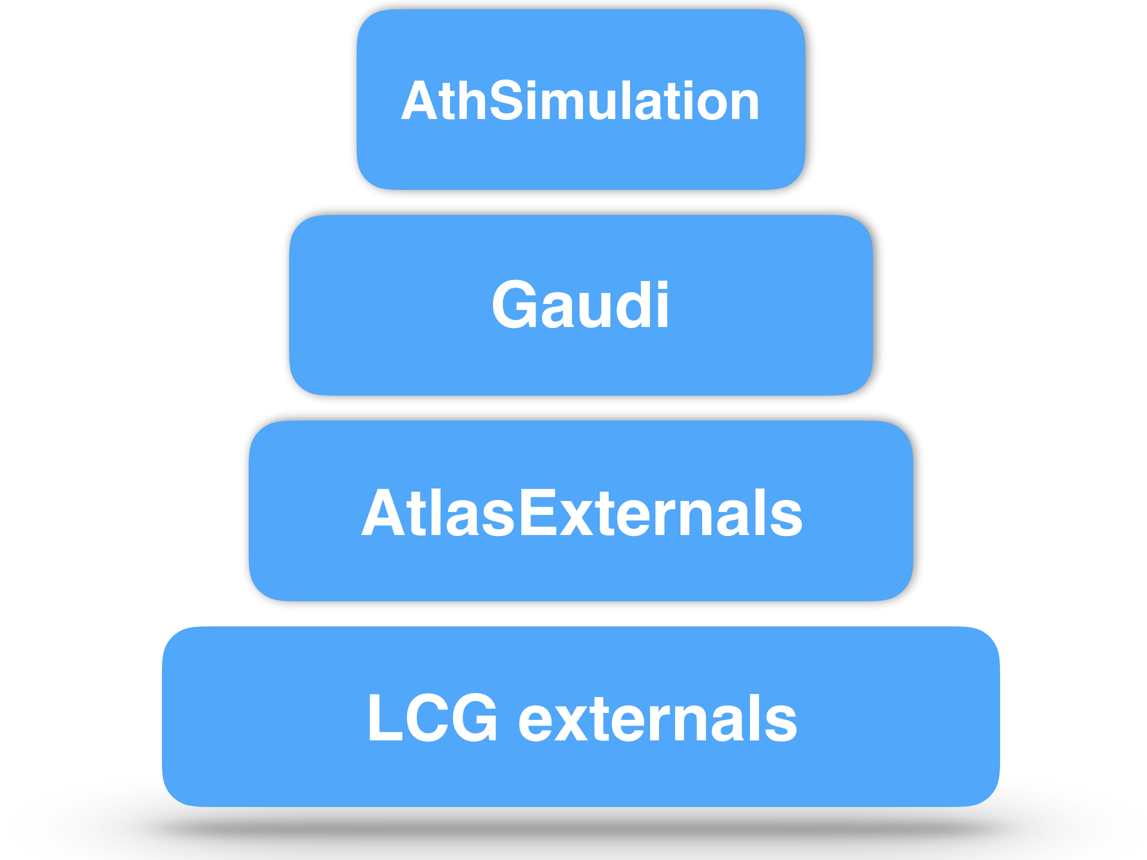
- Growing interest in new/different architectures: Aarch64, POWER8
  - Increasing interest from Industry

Need ATLAS "product" and "benchmark" to test on these new systems

AthSimulation

Enter AthSimulation:

- A fraction of the packages of Athena (~345 compared to ~2400)
- Much quicker compile time
- Potential for errors in port decreases
- Geant4 gives a good CPU load
- Good for simulation and validation
- Implement this in Jenkins
- Build this using CMake - be pioneers



Tools: CMake and Jenkins

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

- CMake: Everyone here should know plenty about this now! (Attila will probably go into more detail on Wednesday)
- Jenkins: A Continuous Integration tool (see Alex's talk on Wednesday)

The very general idea:

- Retrieve code from nightly builds (either tarballs, git etc)
- Tailor to build locally
  - Have tried to make this as general as possible, still some work to do
- Upload to Gitlab repository
- Create Jenkins configuration for project

Everything is self-contained in a Jenkins instance!

Jenkins instance

Self-contained: completely portable[1] requiring no external scripts

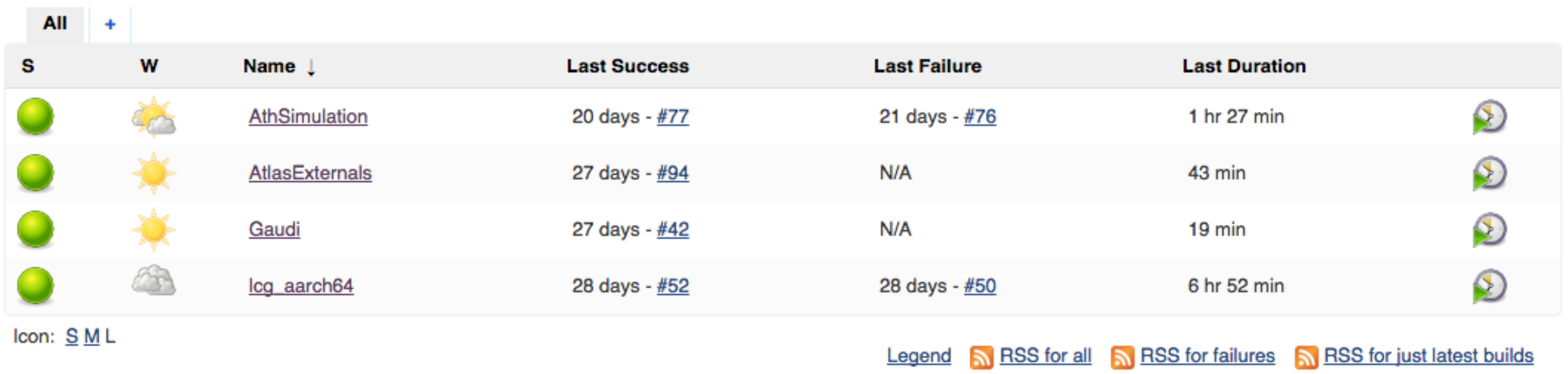
- Transparent: can see exactly what's being called in instance
- Opens up potential to port to many different architectures!
  - Example: On Mac, works until predictable various LCG failures
- No reason we can't scale this current instance up to full Athena builds
- CMake made this whole process orders of magnitude easier
  - Most of the changes for the full Athena build (with cmt) was introducing architecture tags
- From AtlasExternals and up, very few architectural problems/changes (none in AthSimulation[2]!)

[1] There are obviously caveats, which will be explained soon ;)
[2] At least, none that we know of

Jenkins instance

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

- Install Jenkins on desired computer (normally installs it to /var/lib/jenkins or /Users/$USER/.jenkins)
- Should see some xml files, jobs, logs, plugins dir etc. Remove all of them.
- Unpack contents of https://gitlab.cern.ch/jwsmith/Aarch64JenkinsConfiguration [2]
- Start a local server: http://localhost:8080/
- Should see something like: (minus build history)



All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
●	☀	AthSimulation	20 days - #77	21 days - #76	1 hr 27 min	
●	☀	AtlasExternals	27 days - #94	N/A	43 min	
●	☀	Gaudi	27 days - #42	N/A	19 min	
●	☁	lcg_aarch64	28 days - #52	28 days - #50	6 hr 52 min	

Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds

- Go to "lcg_aarch64" (will have to rename if making more general) and select "Build with parameters"
- Sit back and relax…

[2] Provided by SCM Sync configuration plugin which pushes Jenkins configuration changes to repository

AthSimulation: Configuration example

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

- Set some parameters within configuration:

CMAKECONFIG = aarch64-ubuntu1404-gcc49-opt
HEPTOOLS_VERSION = 84
LCG_install
AtlasExternals
GAUDI_INSTALLAREA
ATLAS_EXTERNAL = area for externals (Geant4 ATLAS headers) and “hacked” packages (libunwind)

- Could use CMake plugin, I use execute shell

Jenkins execute shell

```
#!/bin/bash
# Export some paths: this will be cleaned up
export CMAKE_PREFIX_PATH=<…>

# Some more path finding
…

mkdir -p build ; cd build
/home/jwsmith/cmake-3.5.0-install/bin/cmake -DCTEST_USE_LAUNCHERS=TRUE -DCMAKE_INSTALL_PREFIX=/
AthSimulation/1.0.0/InstallArea/${CMAKECONFIG} -DGAUDI_INSTALLAREA=${GAUDI_INSTALLAREA} ../
AthSimulation_source
make -j8
DESTDIR=../install make install
```

Project breakdown

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

LCG packages:

- Are a number of patches/changes
- Have had full build for awhile now
- Currently building from LCG 84

AtlasExternals:

- Most number of hacks due to me not knowing exactly how it's built in NICOS:
  - Several LCG paths hardcoded
- AtlasDeconstruction and Yampl found locally in ~/externals
- AtlasDSFMT: can't build any SSE related libraries

GAUDI:

- Pretty much builds out of the box (v27r1, ATLAS branch)

AthSimulation:

- Slightly customized project root CMakeList.txt file
- Some changes to various package CMakeLists.txt files (should be in trunk now, Graeme?)

```
# Optional dependencies, only used when not in the AthSimulationBase release:
set( extraPackages )
if( NOT SIMULATIONBASE )
  set( extraPackages Calorimeter/CaloEvent )
endif()
```

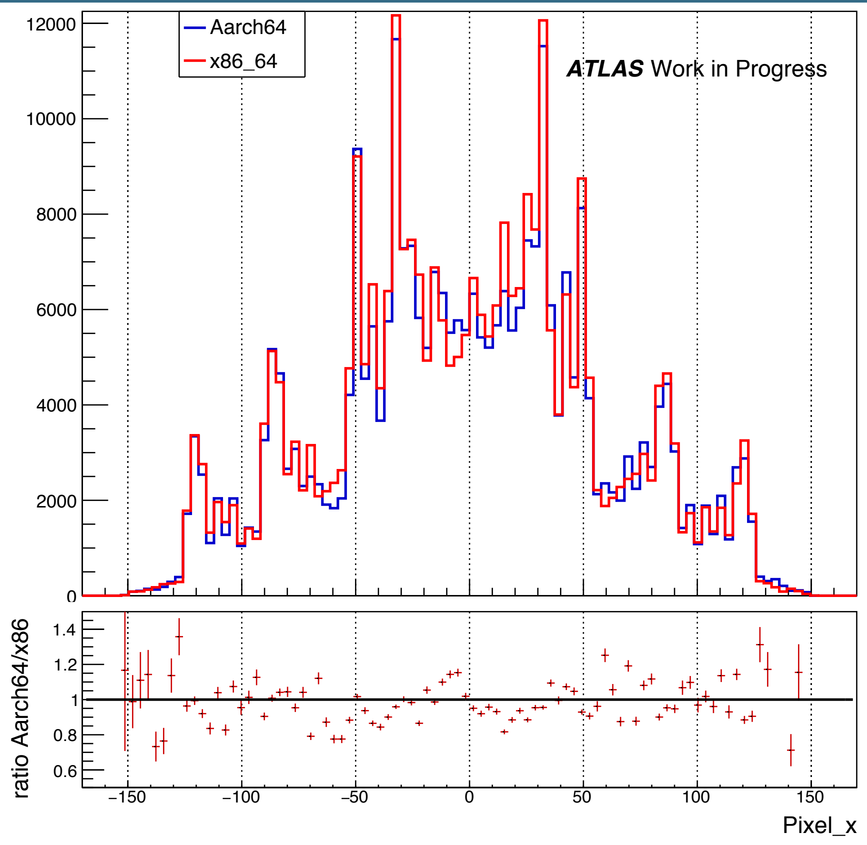
First plots

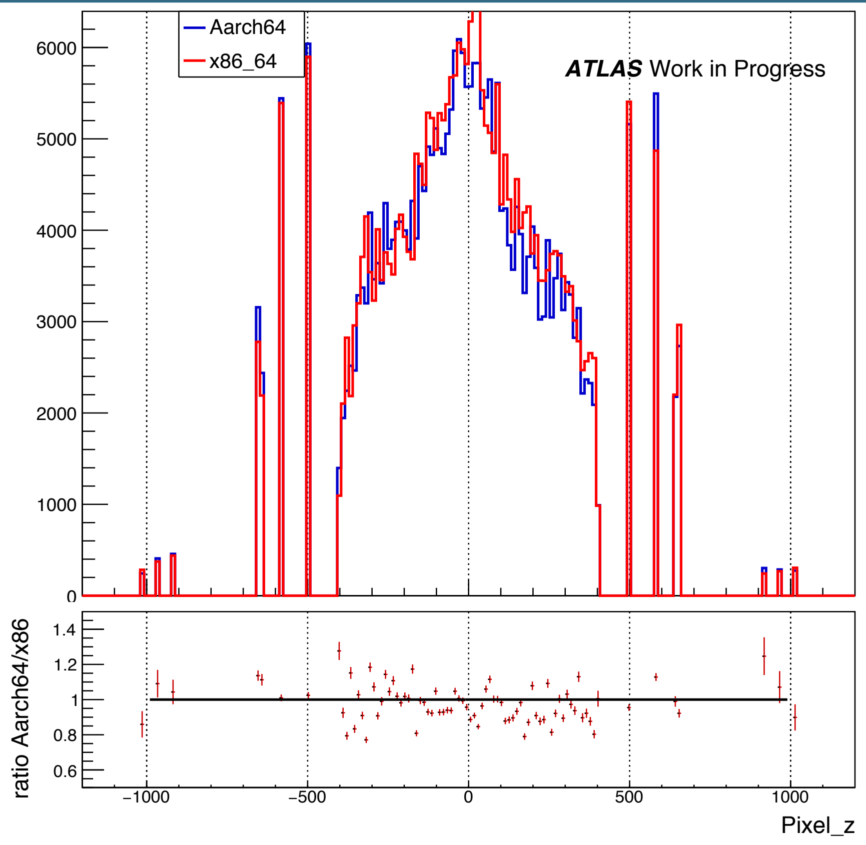
- On each individual architecture (aarch64 and lxplus running nightly release):
  - AtlasG4_tf.py --inputEVNTFile '/afs/cern.ch/atlas/offline/ProdData/16.6.X/16.6.7.Y/ttbar_muplusjets-pythia6-7000.evgen.pool.root' --outputHITSFile 'test.HITS.pool.root' --maxEvents ‘100' --skipEvents '0' --randomSeed '10' --geometryVersion 'ATLAS-R2-2015-03-01-00_VALIDATION' --conditionsTag 'OFLCOND-RUN12-SDR-19' --DataRunNumber '222525' --physicsList 'FTFP_BERT' --postInclude ‘PyJobTransforms/UseFrontier.py'
- On lxplus for each output file:
  - athena.py SiHitAnalysis_topOptions.py
  - A file created on Aarch64 can be run in lxplus
    - Closer to heterogeneous computing!

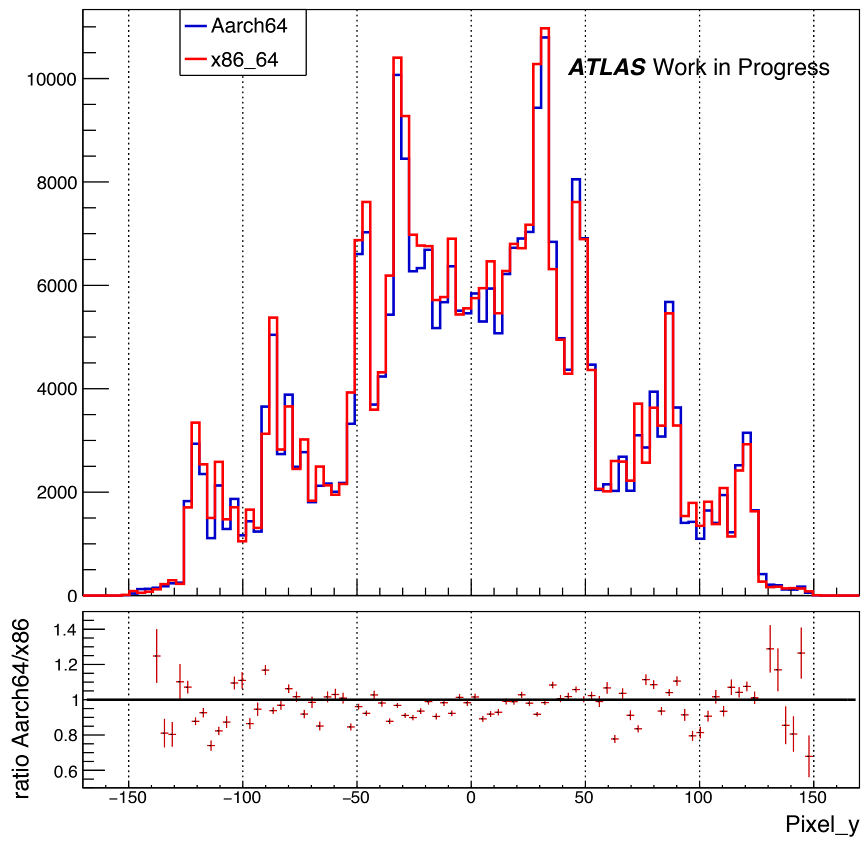
First plots

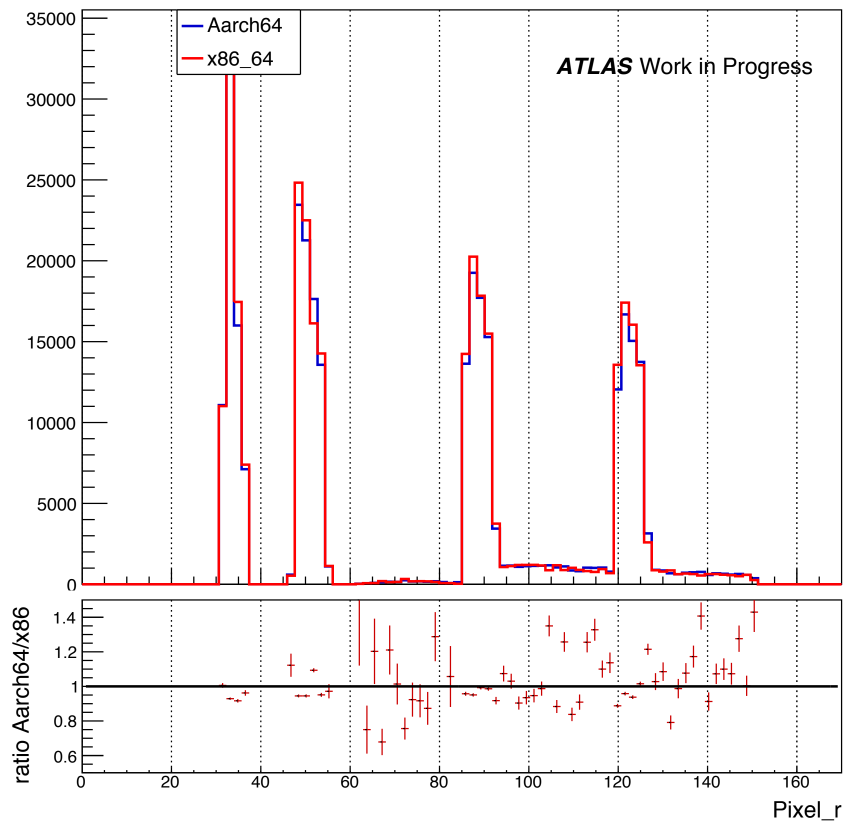
(more interesting plots in backup)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN









Joshua Wyatt Smith

Next steps

While in the same ballpark, this is not a good agreement. Where do the differences stem from?

- 1542 random numbers in respective AtDSFMTGenSvc.out files:
  - 1 single number is different (2nd number in AtlasG4 category if this helps?)
- Try to understand on event by event basis, where is change introduced?
- CVMFS: while this is easy on CentOS 7, less straightforward on Aarch64 Ubuntu
- Wait for new cluster and do this over (we've been "promised" it every week for 6 weeks now)
- Power measurements
- New architectures, POWER8?

Conclusions

- Self-contained Jenkins instance that can be installed on various architectures
  - Almost works out the box, still some “hacks” that need to be addressed
  - Possibility to expand this to full ATLAS codebase
- First output via simulation jobs provides reasonable agreement
  - But, differences look quantitively too large
- Further testing needed, different type of jobs?

Thank you to …

John Chapman,

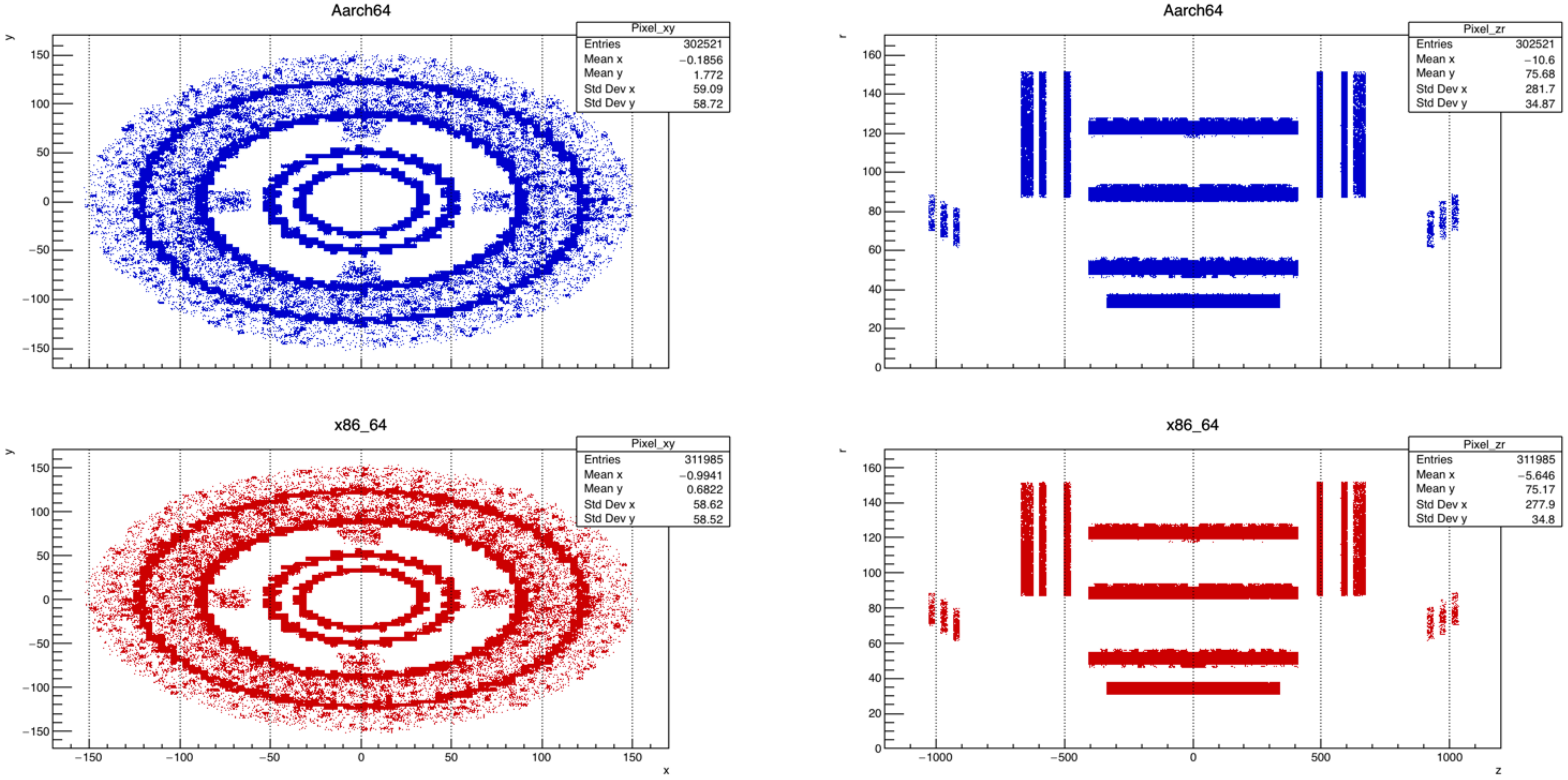
Attila Krasznahorkay,

Zach Marshall,

Rolf Seuster,

Graeme Stewart

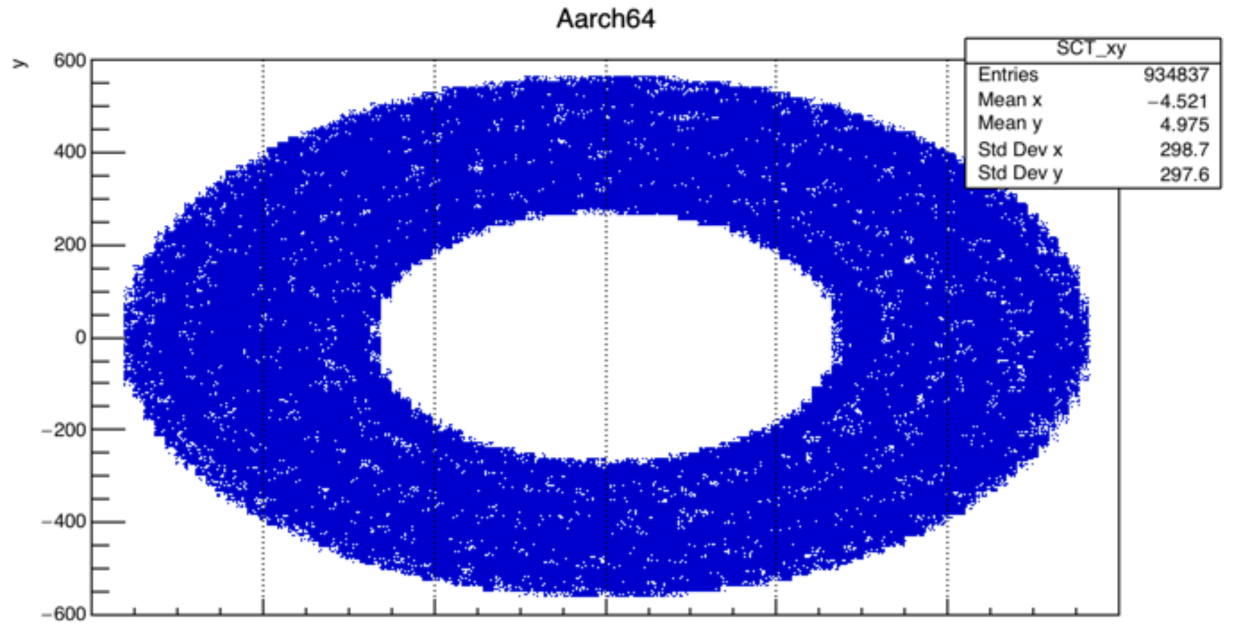
Backup: more plots

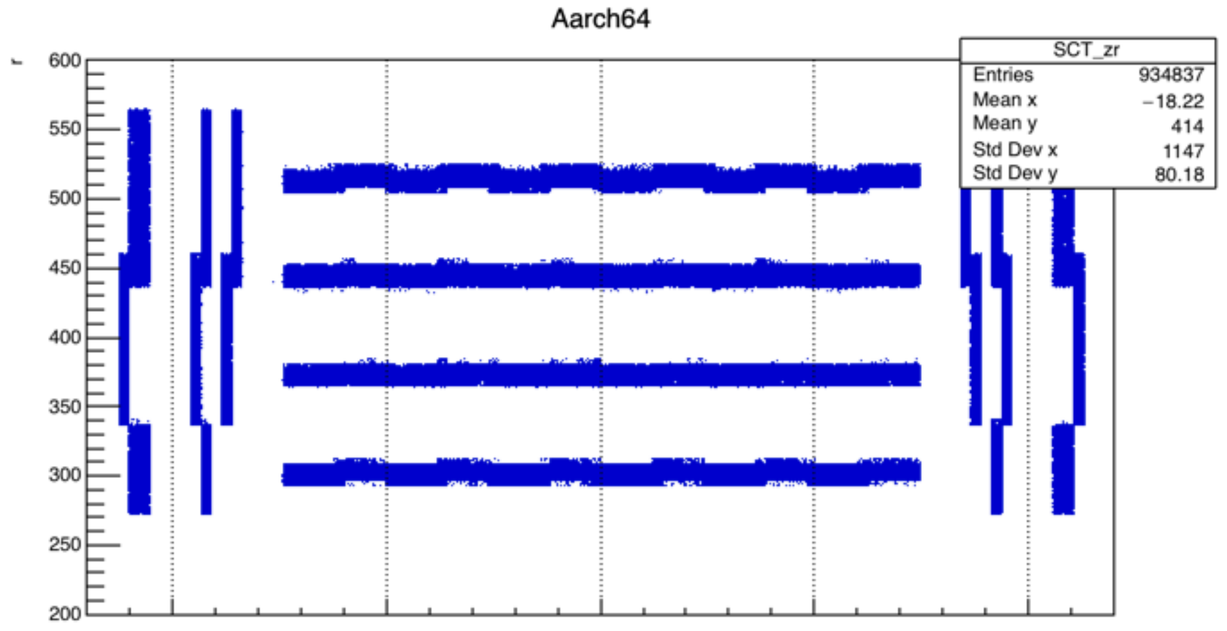


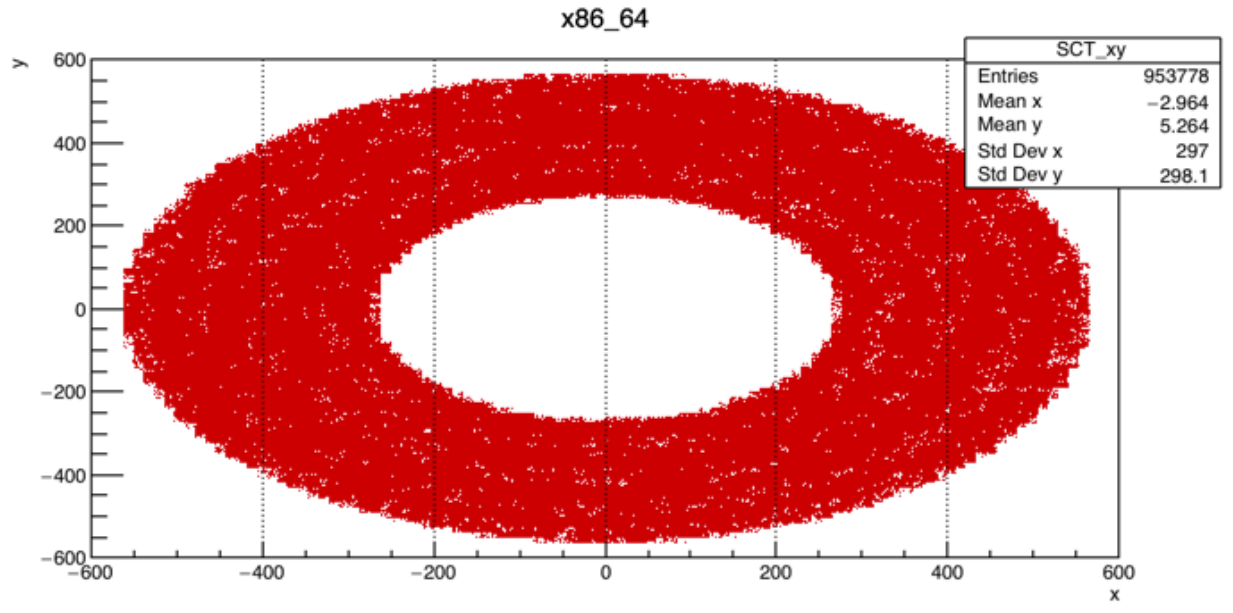
Joshua Wyatt Smith

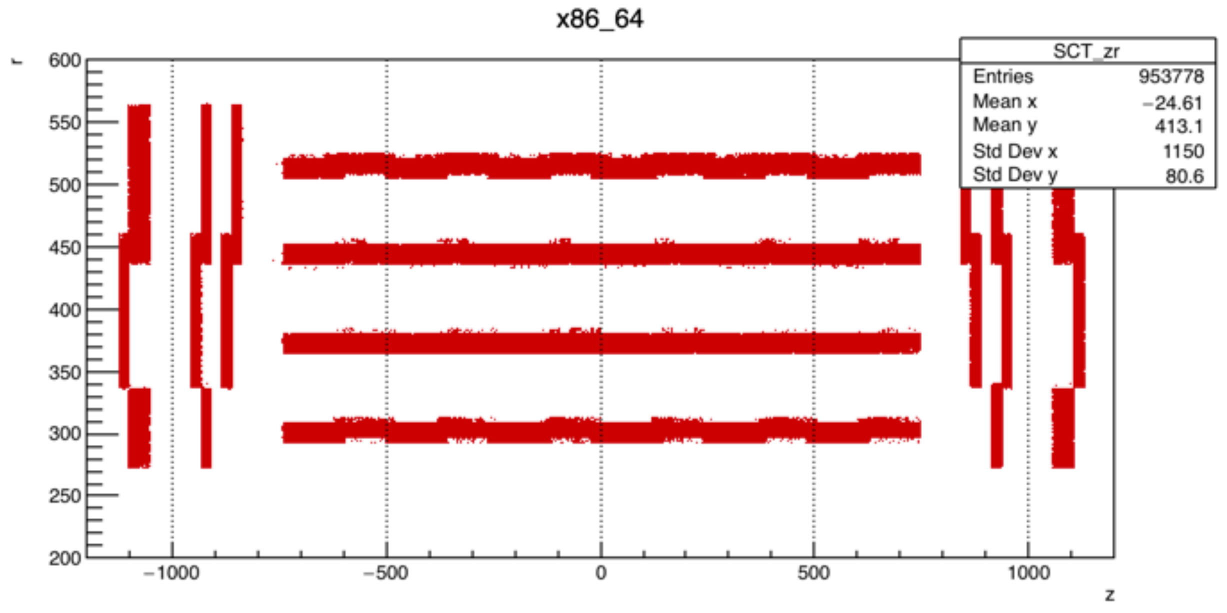
Backup: more plots

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN


Aarch64
SCT_xy
Entries 934837
Mean x −4.521
Mean y 4.975
Std Dev x 298.7
Std Dev y 297.6
y


Aarch64
SCT_zr
Entries 934837
Mean x −18.22
Mean y 414
Std Dev x 1147
Std Dev y 80.18
r


x86_64
SCT_xy
Entries 953778
Mean x −2.964
Mean y 5.264
Std Dev x 297
Std Dev y 298.1
y
x


x86_64
SCT_zr
Entries 953778
Mean x −24.61
Mean y 413.1
Std Dev x 1150
Std Dev y 80.6
r
z