The ALICE Analysis Train System

Markus Zimmermann for the ALICE collaboration

01.09.2014

Motivation

Make analysis efficient on the Grid

Example analysis	Computing demand
• p_T spectrum in PbPb	• 350 users
• 67 TB in 115k files	• 300 analysis per week
• 7500 jobs to run	• 1.6M jobs per week
• Merge histograms	• PBs of data per week

Individual User Analysis → **Centralized System**

Analysis Types

Individual User Analysis

- Can always run
- Each user has to define own dataset
- One analysis per job
- I/O bound
- Job management by hand
- Private bookkeeping

Centralized System

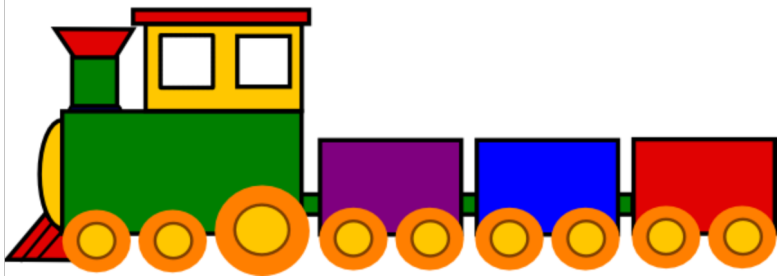
- Typically once a day
- Dataset is defined centrally
- Multiple analysis per job
- CPU bound
- Automatic job management
- Automatic bookkeeping

Objectives

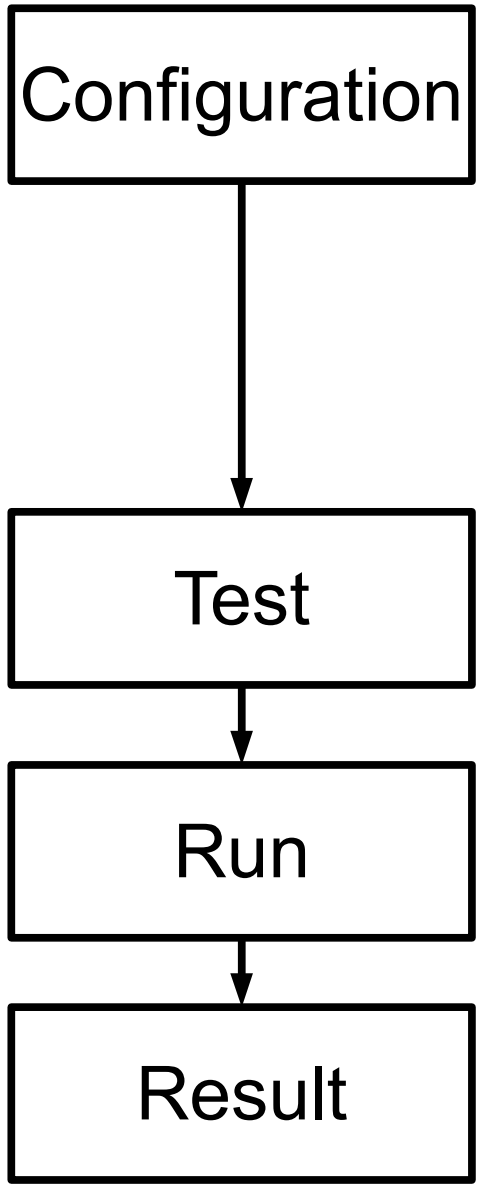
- Hiding Grid complexity
- Optimize resource usage
- Read less often same data
- Automatic bookkeeping

Concept

- Combine several user analysis to a train



- Automatic testing
- Central job management + merging (histograms)
- Train management + bookkeeping in single web page



LEGO[1] Train Configuration

Choose a train

- Working group
- Dataset

defined by user

Parameters

- Software version
- Job configuration

defined by operator

Analysis task

- Analysis code into code repository
  - Wagon is configuration of the code

defined by user

Dataset

- MC/data
- Run numbers

defined by operator

[1] **L**ightweight **E**nvironment for **G**rid **O**perators

Train Test

- Automatic test per wagon on small data sample
  (executed on a dedicated Grid-like machine)

Wagon	Status	Memory consumption		Timing	Merging
		Total	Growth per event		
Base line stdout | stderr	**OK**	250 MB	0.182 KB/evt	34.10ms/evt	**No output**
wagon 1 stdout | stderr	**OK**	591 MB	**0.158 MB/evt**	1.02ms/evt	**OK** merge dir
wagon 2 stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**
Full train stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**

- Operator removes failed wagons

Train Test

- Automatic test per wagon on small data sample
  (executed on a dedicated Grid-like machine)

Wagon	Status	Memory consumption		Timing	Merging
		Total	Growth per event		
Base line stdout | stderr	**OK**	250 MB	0.182 KB/evt	34.10ms/evt	**No output**
wagon 1 stdout | stderr	**OK**	591 MB	**0.158 MB/evt**	1.02ms/evt	**OK** merge dir
wagon 2 stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**
Full train stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**

- Operator removes failed wagons

Train Test

- Automatic test per wagon on small data sample
  (executed on a dedicated Grid-like machine)

Wagon	Status	Memory consumption		Timing	Merging
		Total	Growth per event		
Base line stdout | stderr	**OK**	250 MB	0.182 KB/evt	34.10ms/evt	**No output**
wagon 1 stdout | stderr	**OK**	591 MB	**0.158 MB/evt**	1.02ms/evt	**OK** merge dir
wagon 2 stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**
Full train stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**

- Operator removes failed wagons

Train Test

- Automatic test per wagon on small data sample
  (executed on a dedicated Grid-like machine)

Wagon	Status	Memory consumption		Timing	Merging
		Total	Growth per event		
Base line stdout | stderr	**OK**	250 MB	0.182 KB/evt	34.10ms/evt	**No output**
wagon 1 stdout | stderr	**OK**	591 MB	**0.158 MB/evt**	1.02ms/evt	**OK** merge dir
wagon 2 stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**
Full train stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**

- Operator removes failed wagons

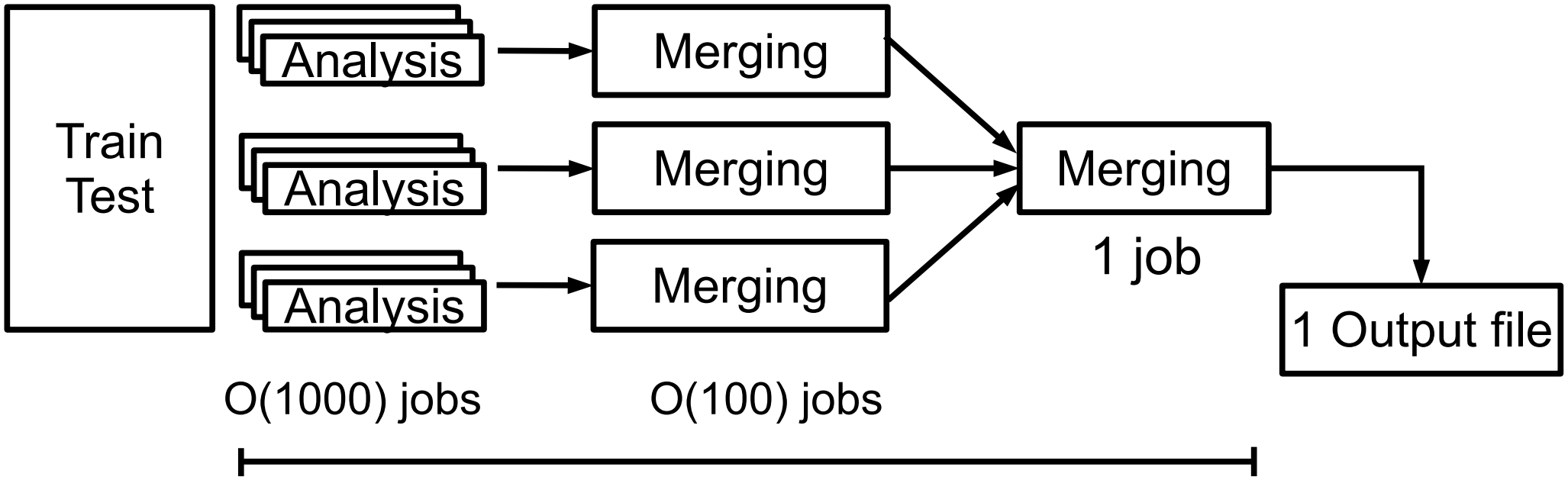
Train Test

- Automatic test per wagon on small data sample
  (executed on a dedicated Grid-like machine)

Wagon	Status	Memory consumption		Timing	Merging
		Total	Growth per event		
Base line stdout | stderr	**OK**	250 MB	0.182 KB/evt	34.10ms/evt	**No output**
wagon 1 stdout | stderr	**OK**	591 MB	**0.158 MB/evt**	1.02ms/evt	**OK** merge dir
wagon 2 stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**
Full train stdout | stderr	**Failed**		/evt	ms/evt	**Not tested**

- Operator removes failed wagons

Train Run



This turn around time has to be “short”

- Turn around time in
  - 2012: 49 hours
  - 2014: 14 hours

Bookkeeping

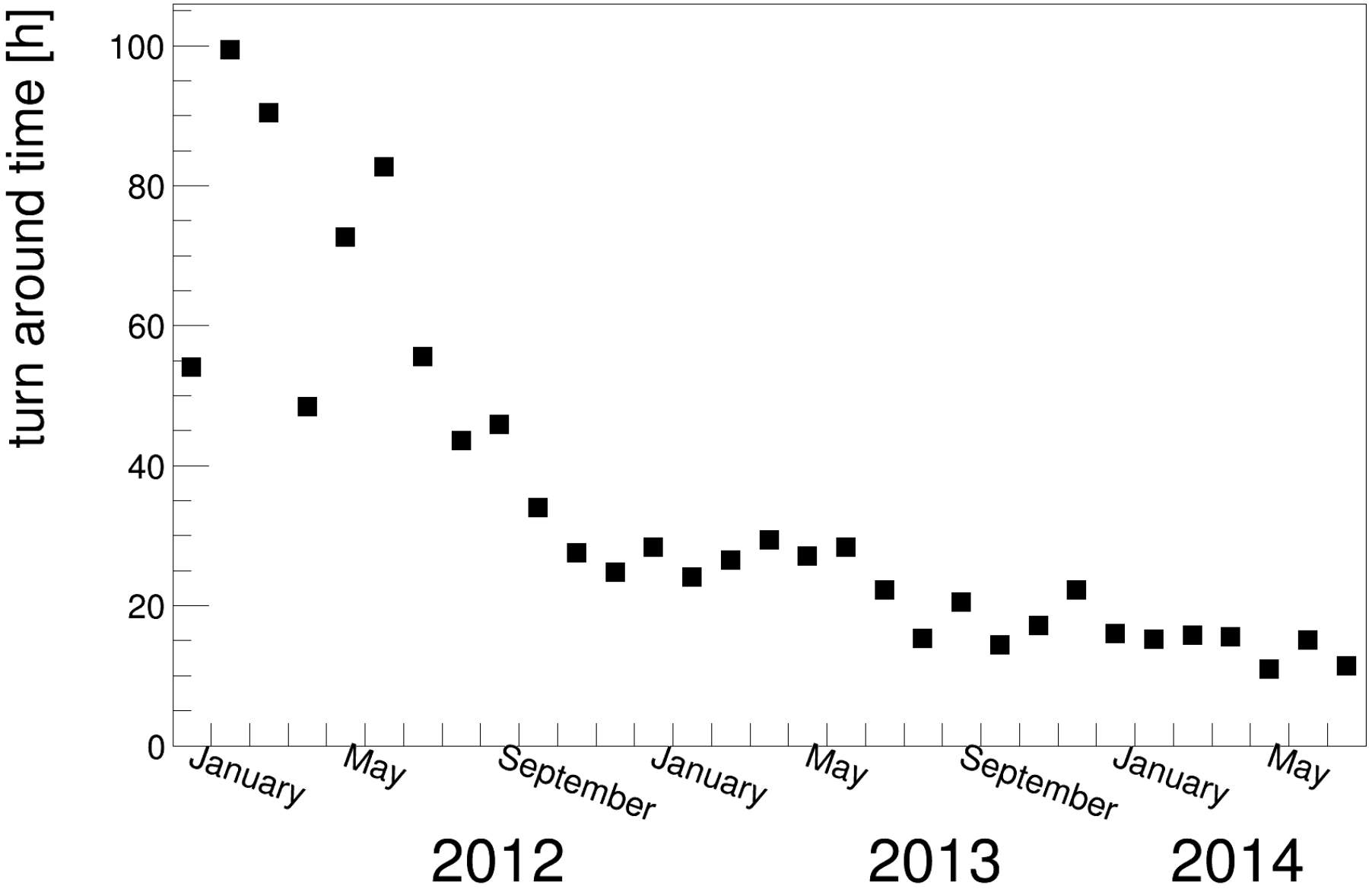
- Bookkeeping is identical for all train runs
- Train run files are accessible with a browser
  - Train configuration and result file are archived
  - Available to everyone inside ALICE
  - Long time data preservation

Usage Statistics

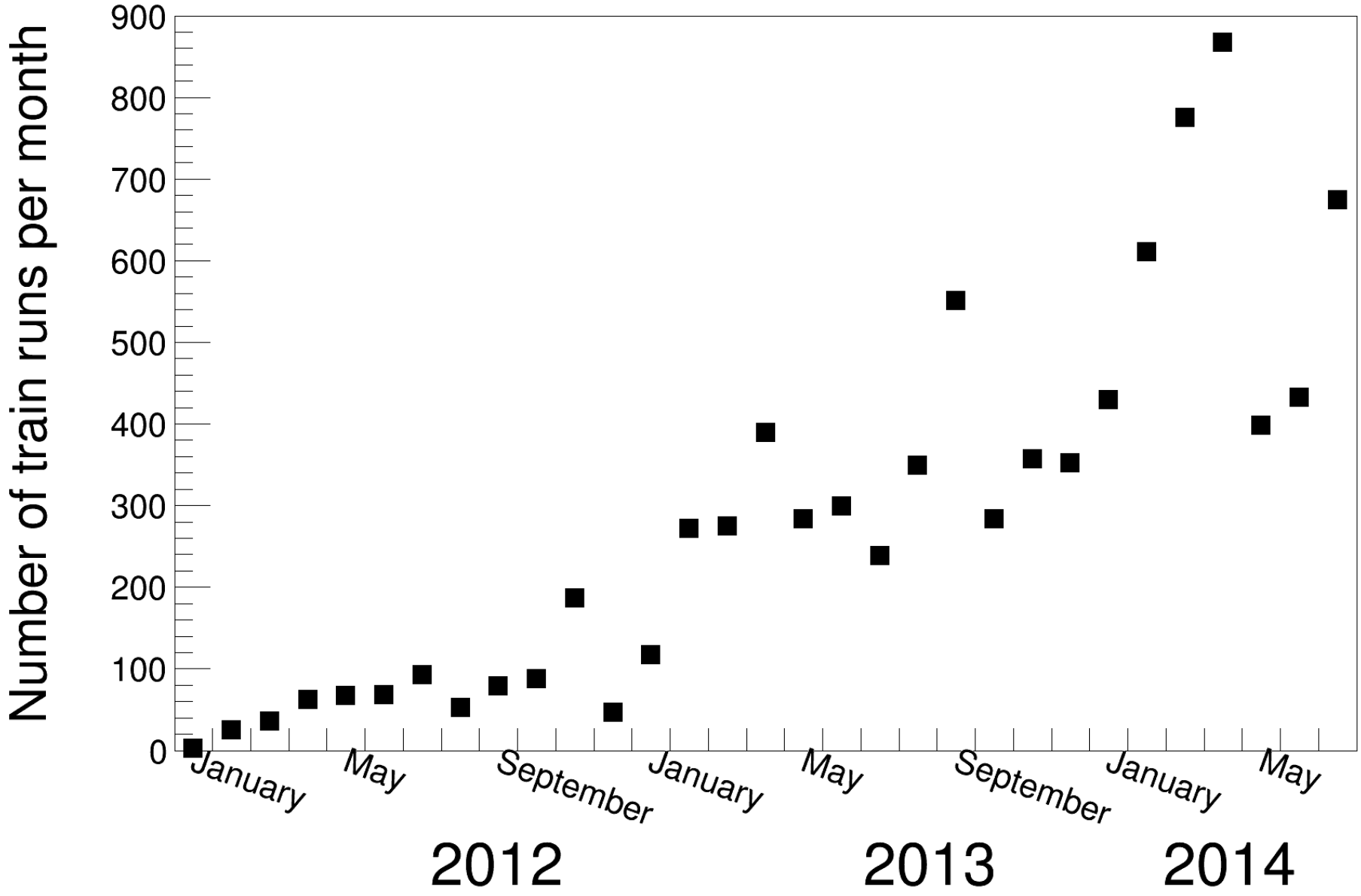
Statistics

Train Status	2012	2013	2014 (extrapolated)
Users	60	127	188
Trains	42	69	79
Train runs	1537	4794	7446
Number of jobs	12 million	26 million	36 million
Train wagons/run	14.9	10.1	8.9
Part of the user analysis done with the trains	27%	57%	70%
Processed data	-	75 PB	130 PB
Turn around time	49 hours	22 hours	14 hours

Turn Around Time



Number of Train Runs



How to Convince Users

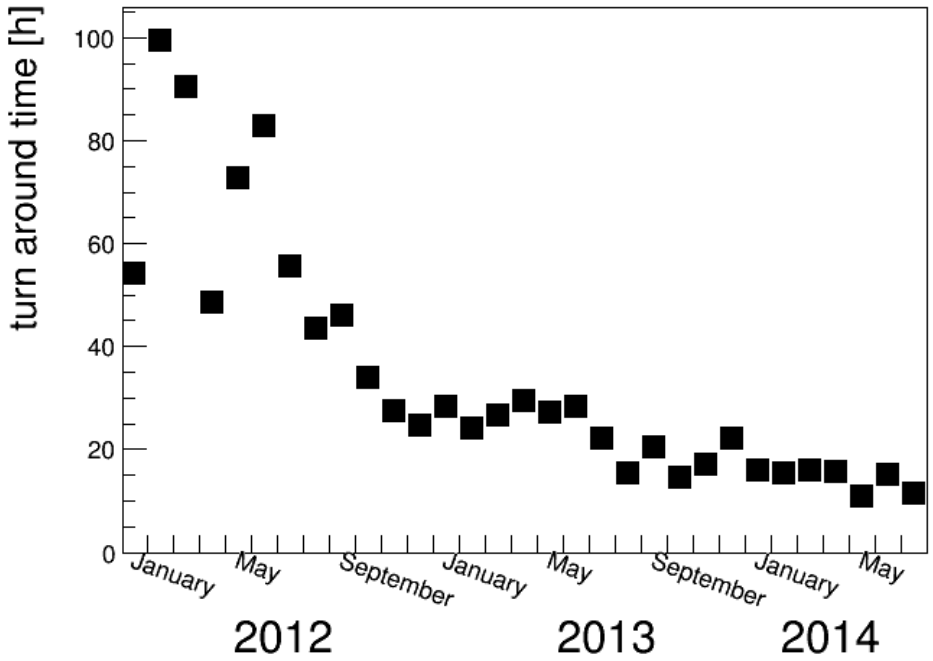
- Users save time
  - Wagon setup is simpler than submitting analysis jobs
  - Easier to learn than the Grid job management
- Well defined job management
  - Train jobs are more stable than individual jobs
  - Grid support finds bugs easier → faster support
- Obtaining results fast enough

How to Convince Users

- Users save time
  - Wagon setup is simpler than submitting analysis jobs
  - Easier to learn than the Grid job management
- Well defined job management
  - Train jobs are more stable than individual jobs
  - Grid support finds bugs easier → faster support
- Obtaining results fast enough

Win - Win – Situation for users and Grid support

Improvements to Reduce the Turn Around Time

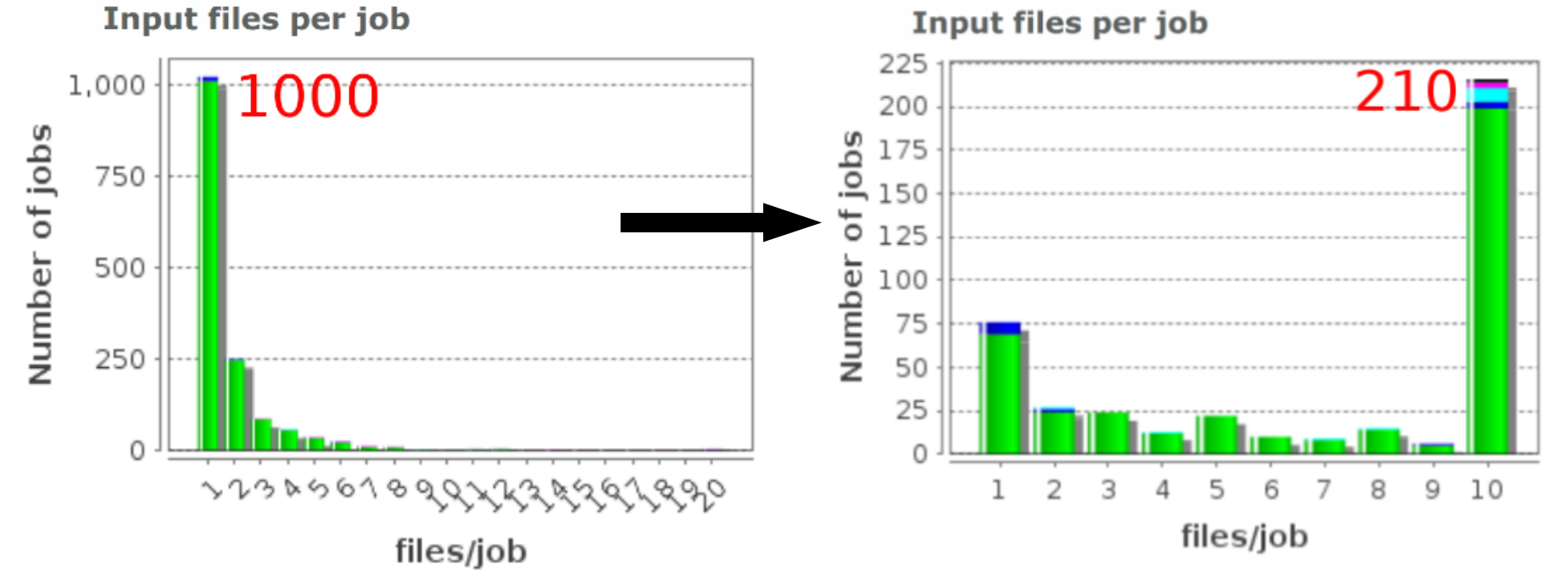


Job Management

- Jobs run on sites which have the data locally
  - As per ALICE computing model
- Read data remotely if train is almost finished
  - Activated if >90% of the jobs successfully finished
  - Speeds up last analysis jobs
- Resubmit jobs automatically if they fail
  - Analysis jobs are resubmitted once
  - Merging jobs are resubmitted up to 3 times

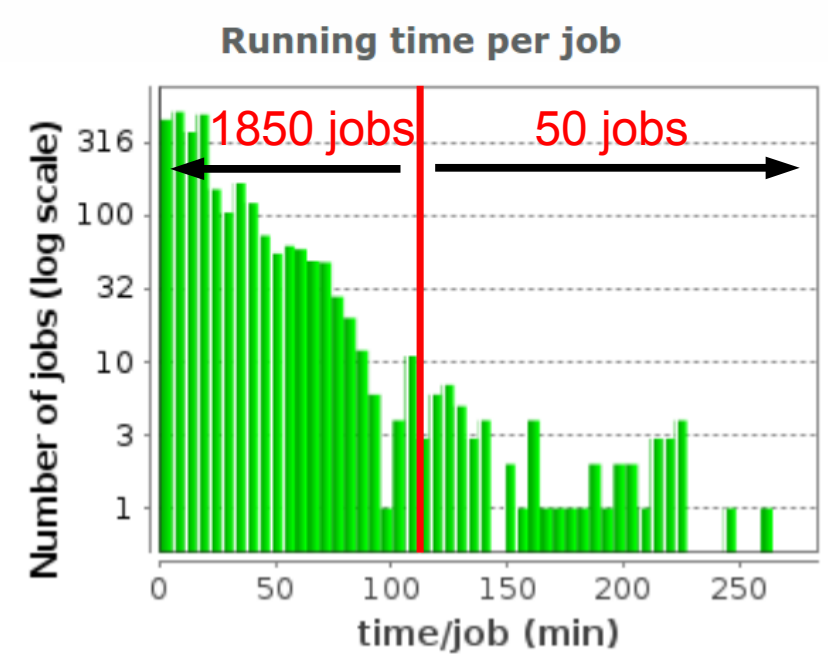
Consolidation of the Datasets

- Files are spread over many storage elements (SE)
- Consolidate file location to maximize files per job
- Significantly reduces number of jobs



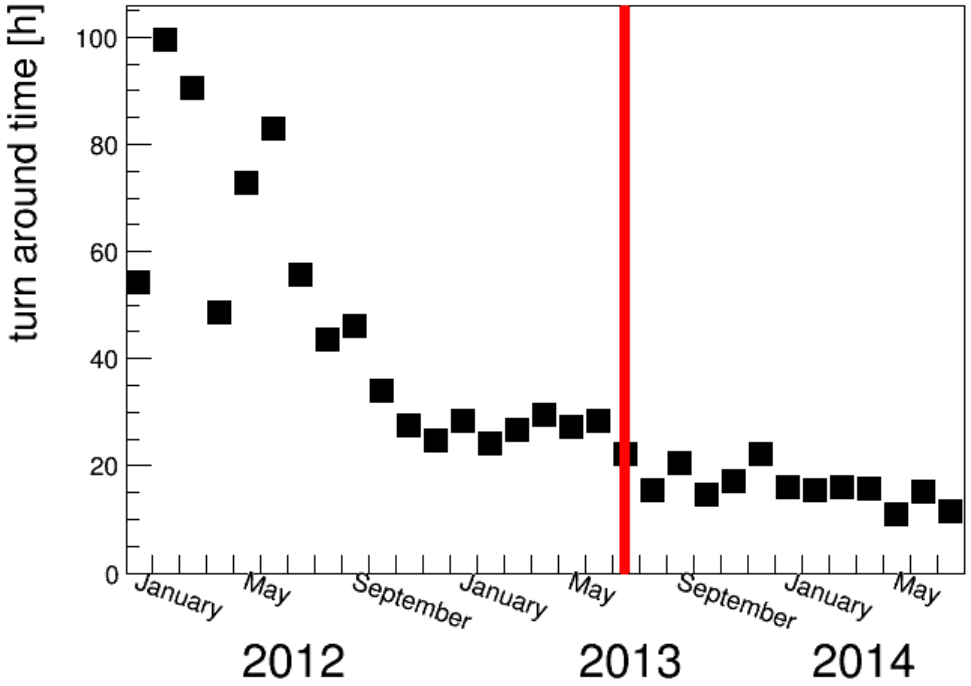
Run Time of the Analysis Jobs

- Running time per job varies very much
  - Most of the jobs are fast
  - Few jobs need significantly longer
- Kill last 2% of the jobs to finish the train earlier
  - Trade statistics for turn around time
  - Can be deactivated on request



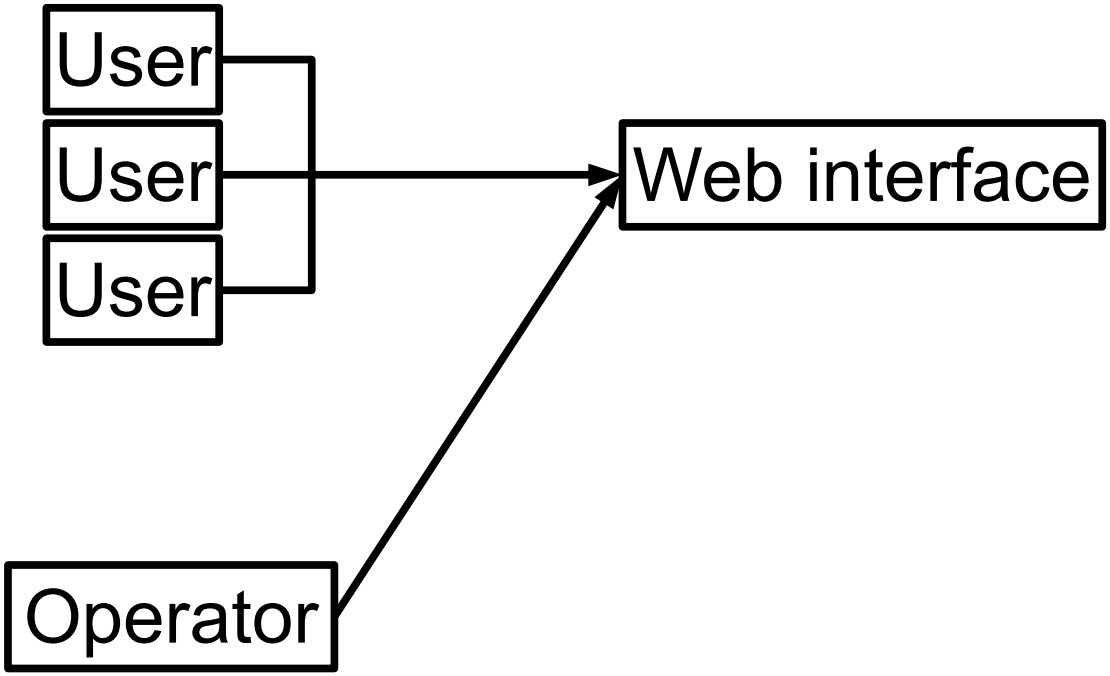
Run Time of the Analysis Jobs

- Running time per job varies very much
  - Most of the jobs are fast
  - Few jobs need significantly longer
- Kill last 2% of the jobs to finish the train earlier
  - Trade statistics for turn around time
  - Can be deactivated on request



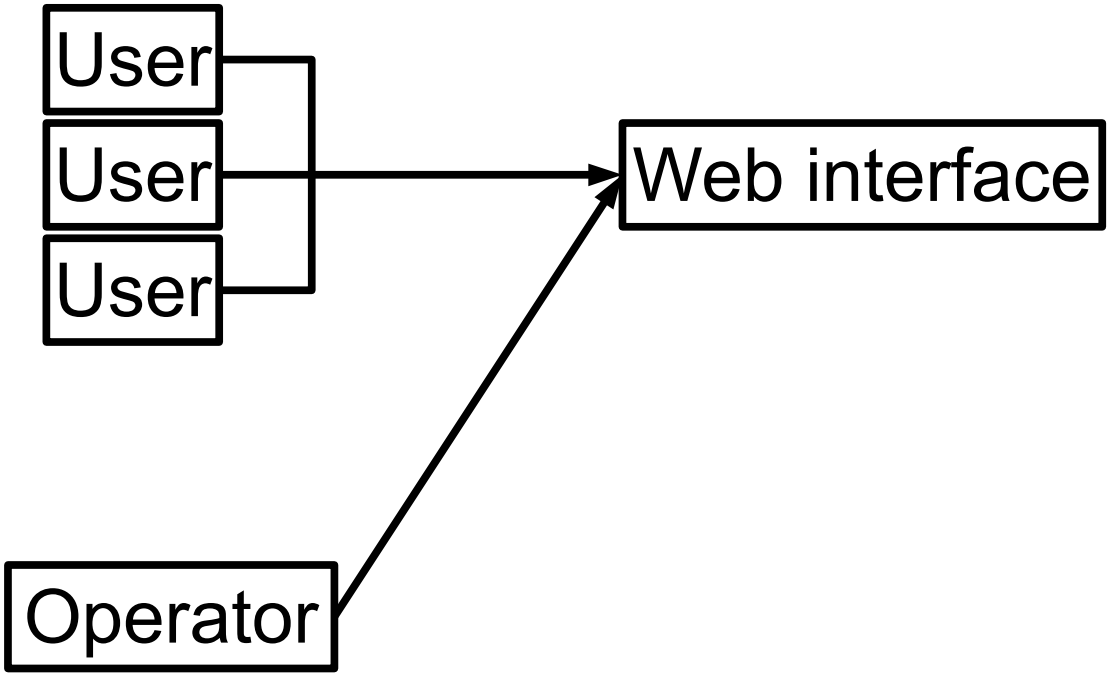
Technical Details

Workflow

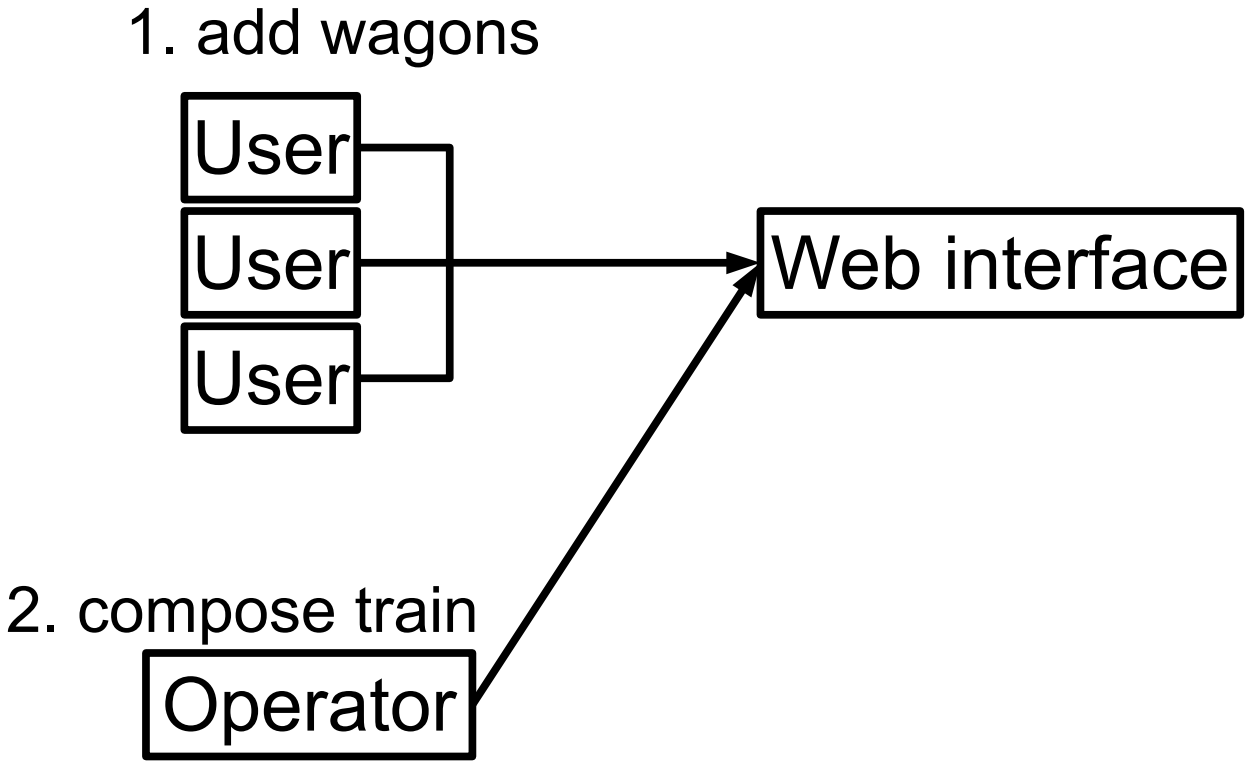


Workflow

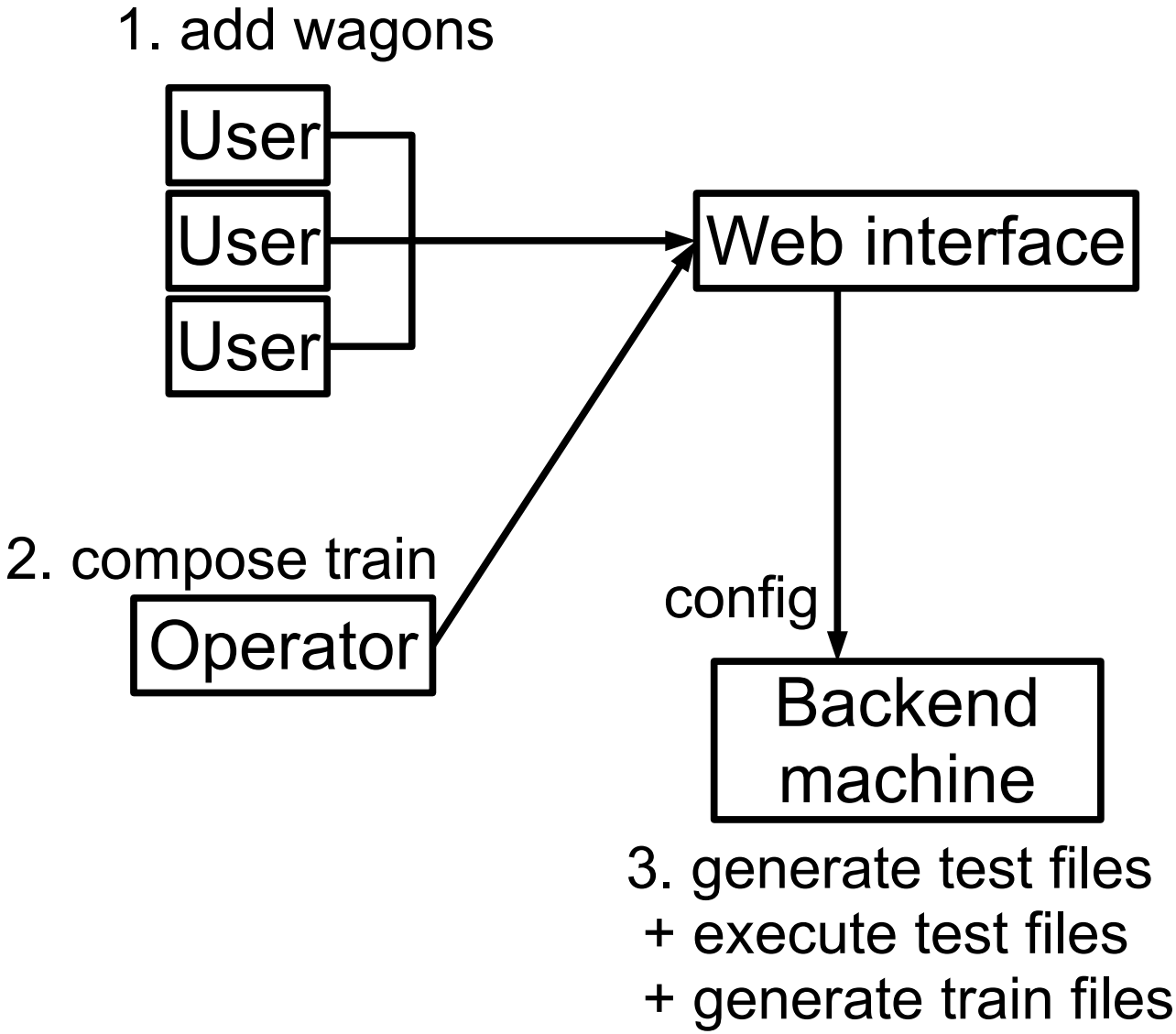
1. add wagons



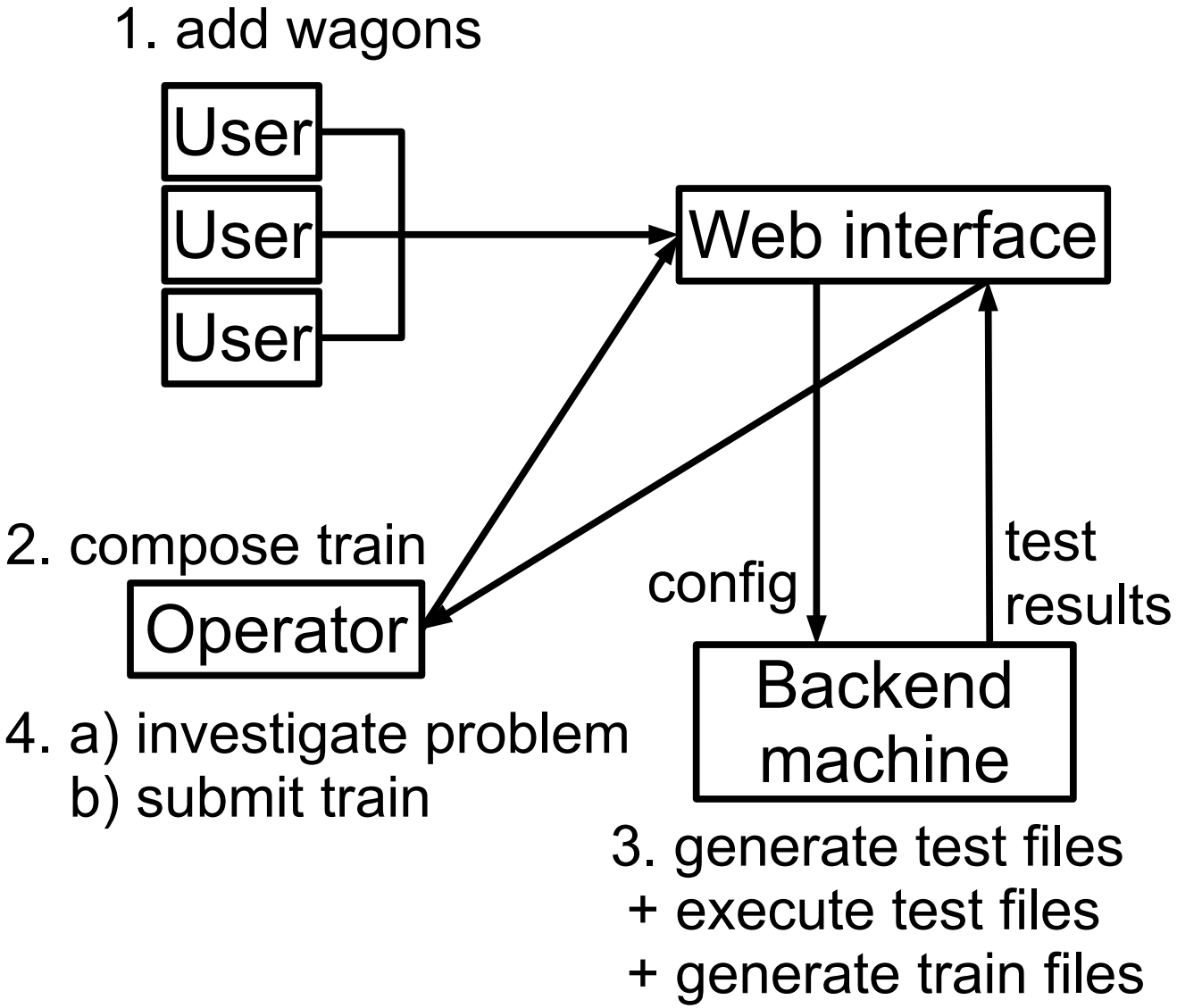
Workflow


1. add wagons
User
User
User
Web interface
2. compose train
Operator

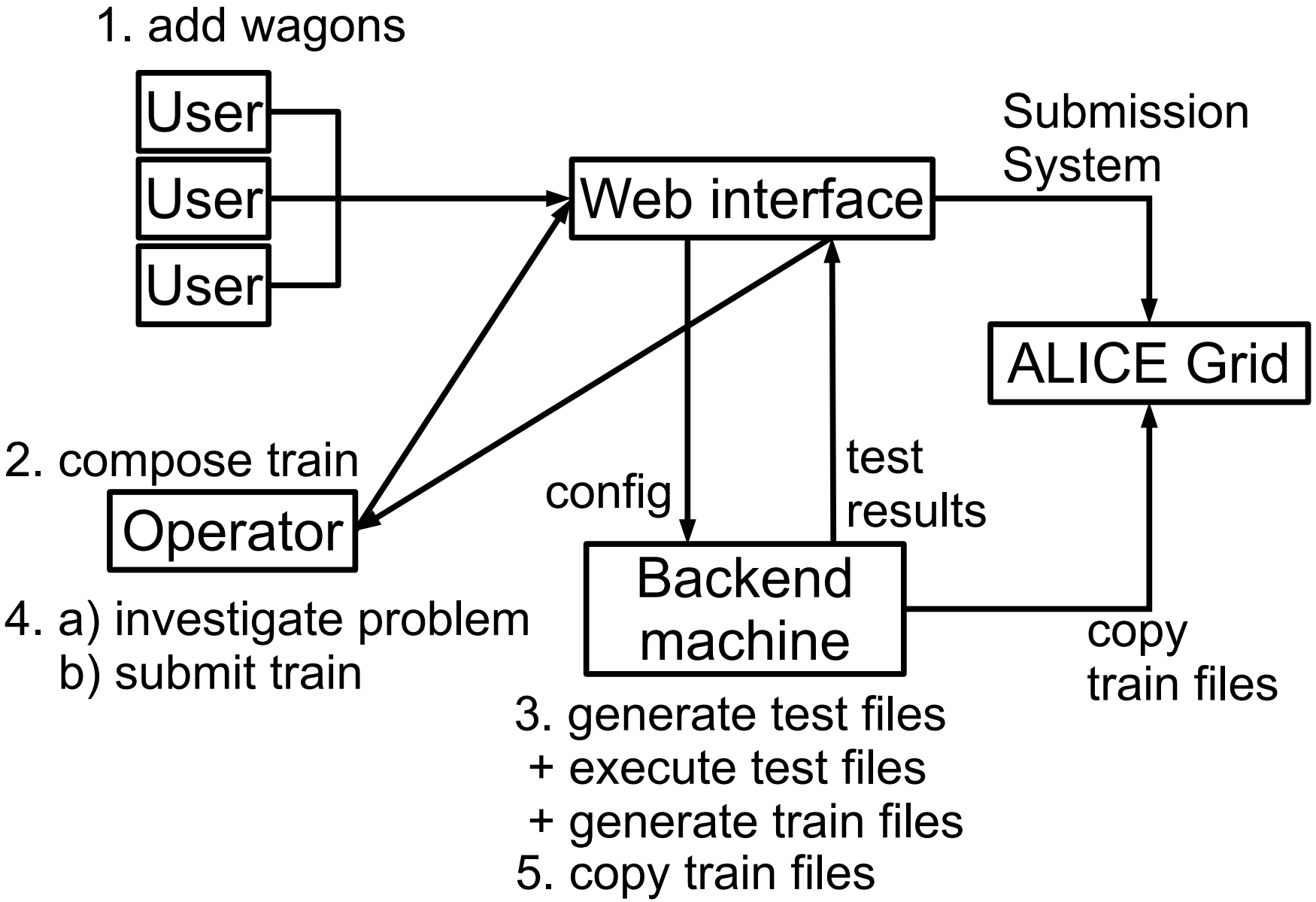
Workflow


1. add wagons
User
User
User
Web interface
2. compose train
Operator
config
Backend machine
3. generate test files
+ execute test files
+ generate train files

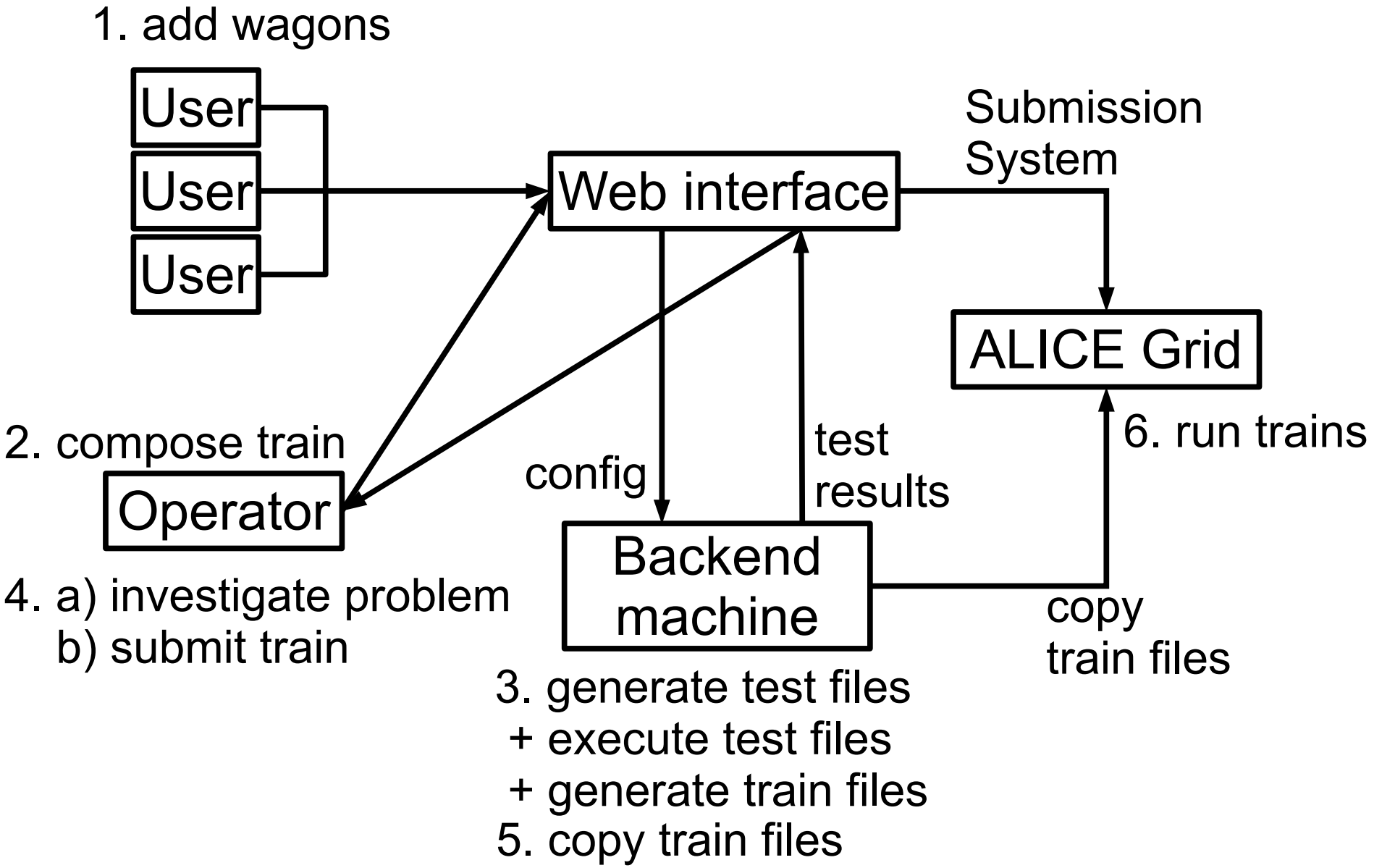
Workflow


1. add wagons
User
User
User
Web interface
2. compose train
Operator
config
test
results
Backend
machine
4. a) investigate problem
b) submit train
3. generate test files
+ execute test files
+ generate train files

Workflow


1. add wagons
User
User
User
Web interface
Submission System
ALICE Grid
2. compose train
Operator
config
test results
Backend machine
copy train files
4. a) investigate problem
b) submit train
3. generate test files
+ execute test files
+ generate train files
5. copy train files

Workflow


1. add wagons
User
User
User
Web interface
Submission System
ALICE Grid
6. run trains
2. compose train
Operator
config
test results
Backend machine
copy train files
4. a) investigate problem
b) submit train
3. generate test files
+ execute test files
+ generate train files
5. copy train files

Summary

- The LEGO trains are a workflow for organized analysis in ALICE
  - 70% of the analysis activity is on trains
- Hiding Grid complexity from the users
- Return the merged output file
- Combine multiple analysis in one Grid job
- Saves time for users and operators
- Improved computing resource management and usage