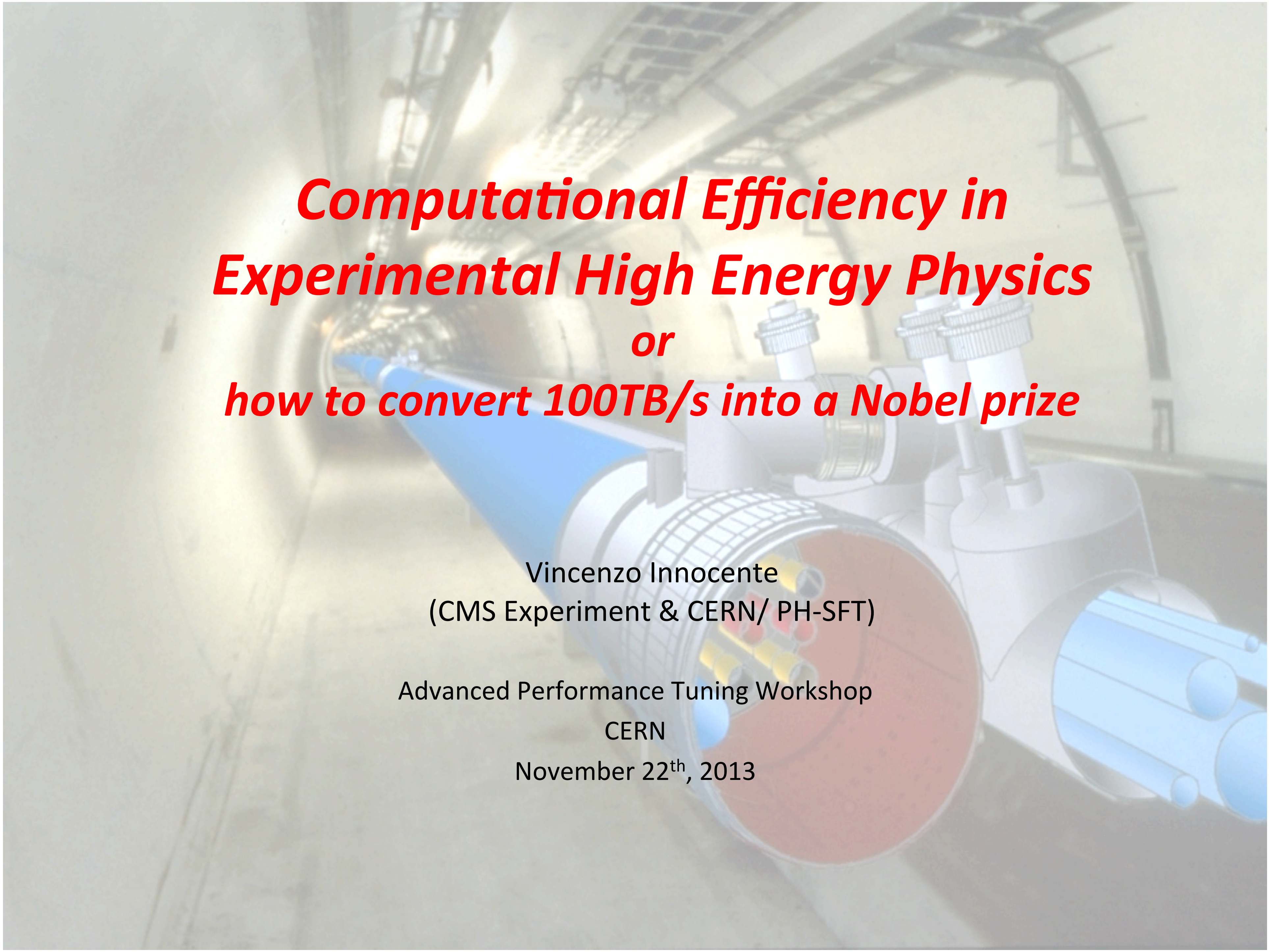


Computational Efficiency in Experimental High Energy Physics
or
how to convert 100TB/s into a Nobel prize

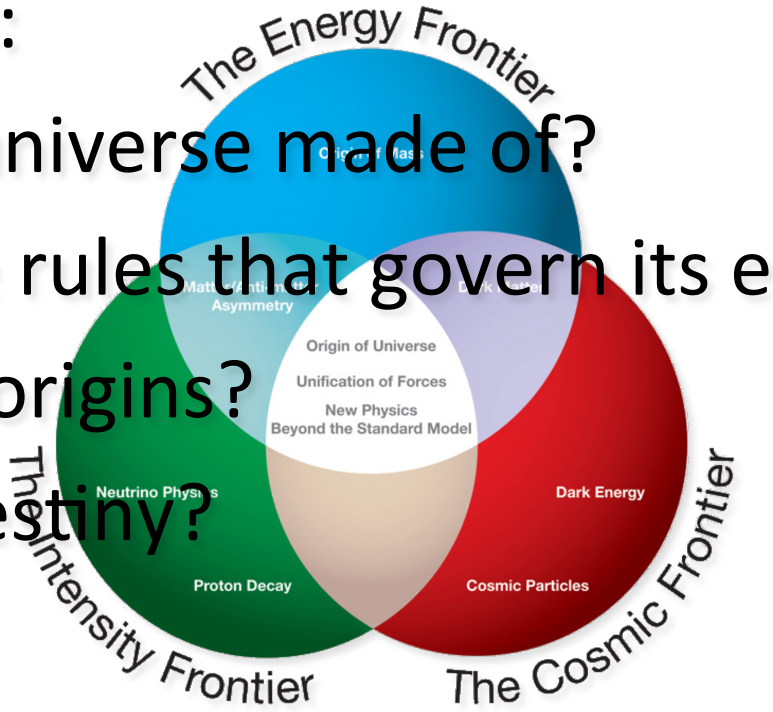
Vincenzo Innocente
(CMS Experiment & CERN/ PH-SFT)

Advanced Performance Tuning Workshop
CERN
November 22th, 2013

Particle Physics Frontiers

Only by exploring these 3 frontiers we can find the answers to the most fundamental questions of the mankind:

- What is the universe made of?
- What are the rules that govern its evolution?
- What are its origins?
- What is its destiny?



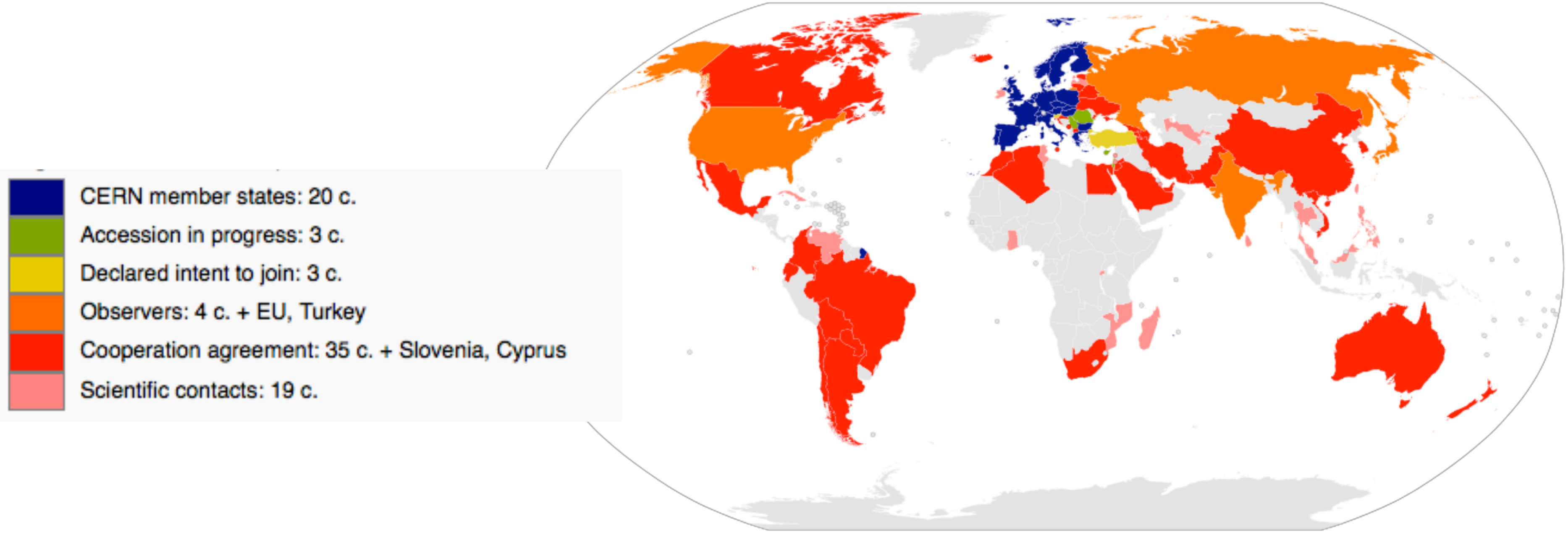
CERN

European Organization for Nuclear Research

Founded in 1954 by 12 countries.

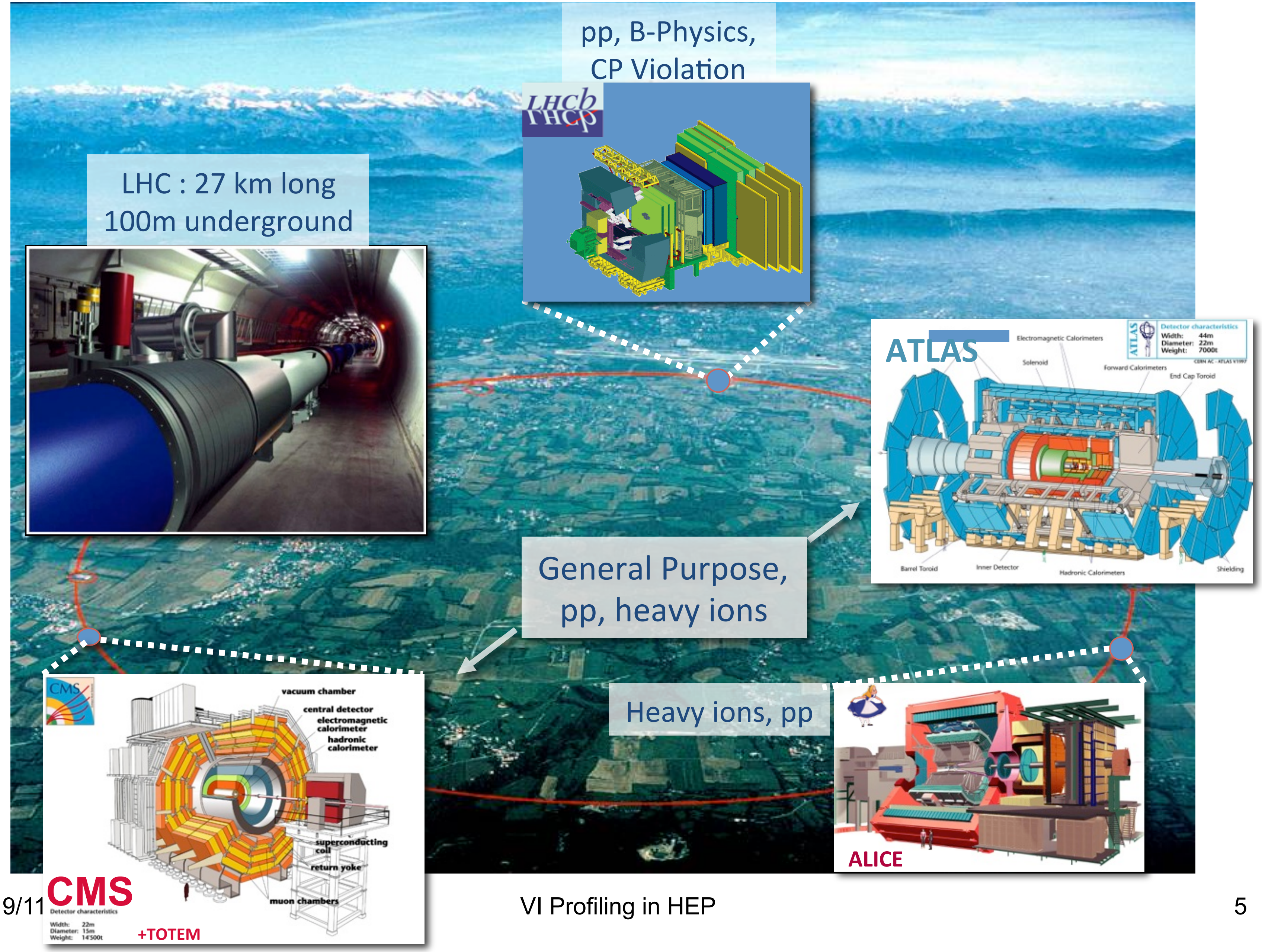
2012: 20 member states

More than 10,000 users all around the world


🟦 CERN member states: 20 c.
🟩 Accession in progress: 3 c.
🟨 Declared intent to join: 3 c.
🟧 Observers: 4 c. + EU, Turkey
🟥 Cooperation agreement: 35 c. + Slovenia, Cyprus
Scientific contacts: 19 c.

DATA PROCESSING IN HEP (LHC)

The Large Hardon Collider at CERN



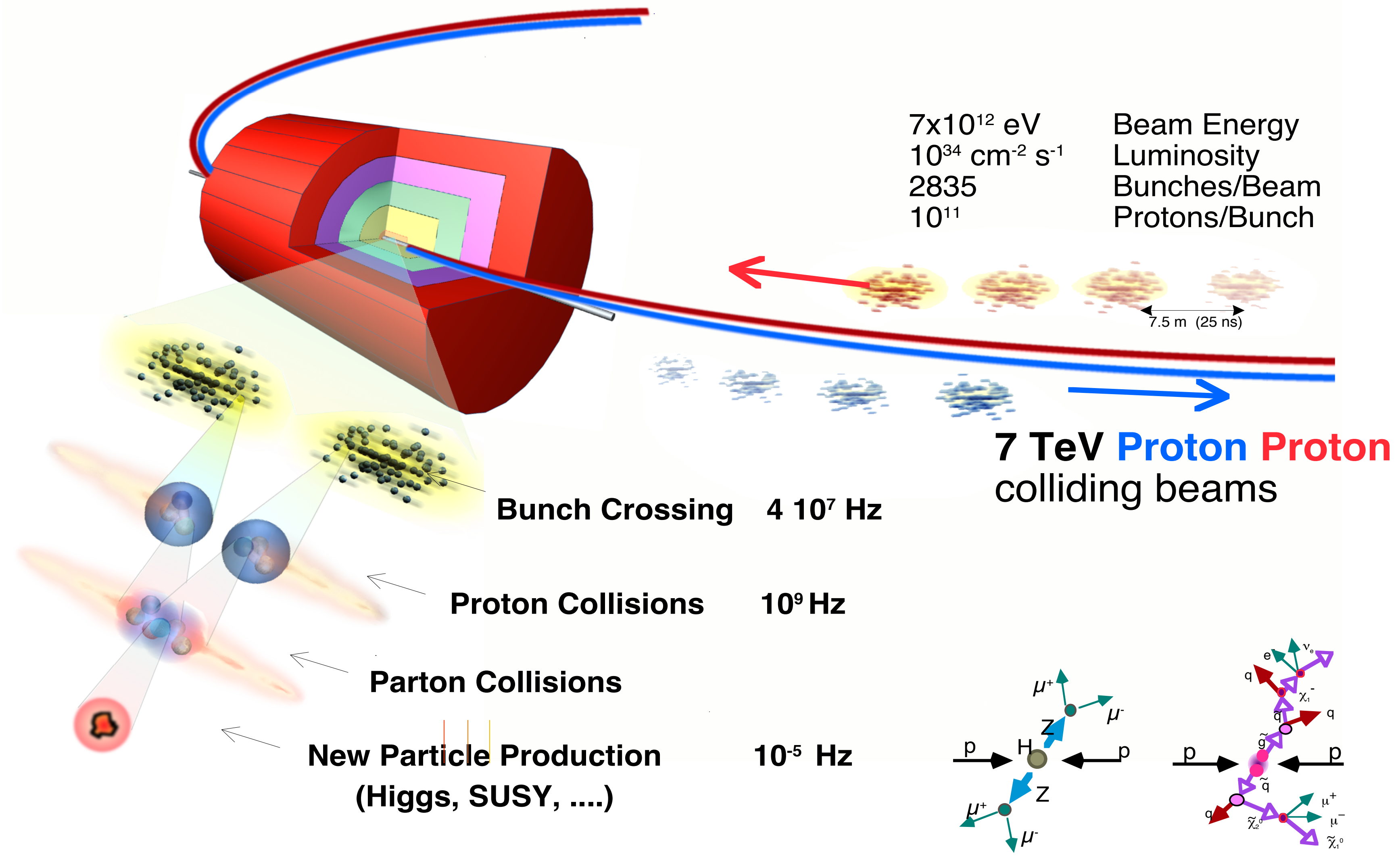
pp, B-Physics, CP Violation

LHC : 27 km long 100m underground

General Purpose, pp, heavy ions

Heavy ions, pp

Collisions at the LHC: summary



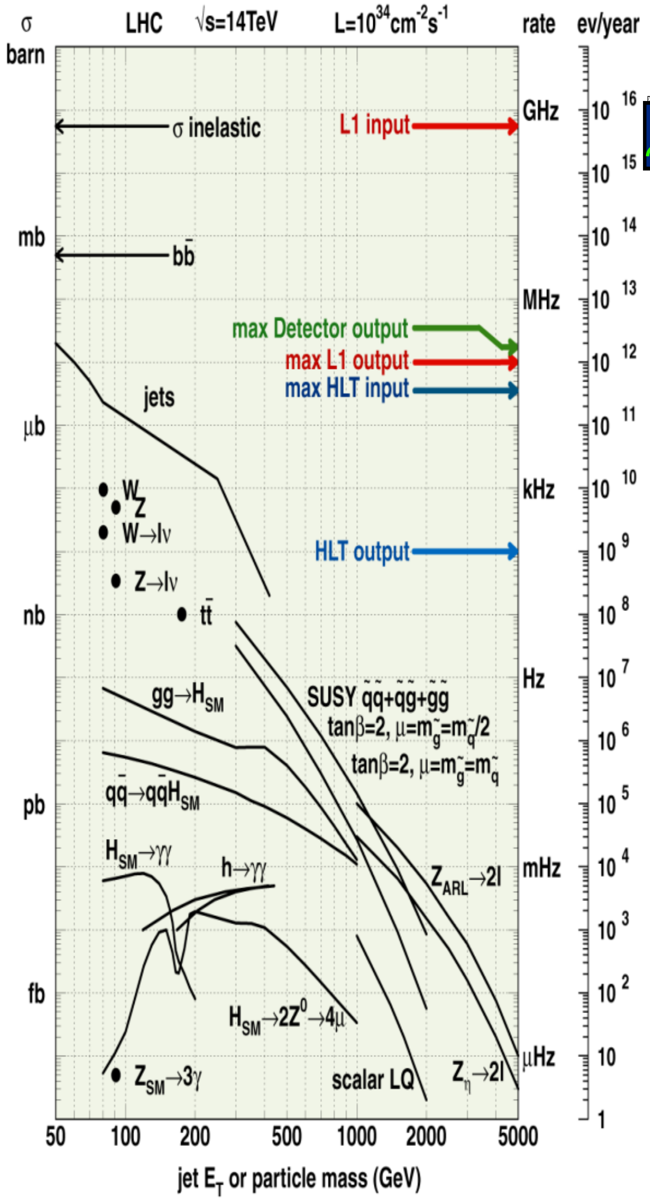
7×10^{12} eV	Beam Energy
10^{34} cm^{-2} s^{-1}	Luminosity
2835	Bunches/Beam
10^{11}	Protons/Bunch

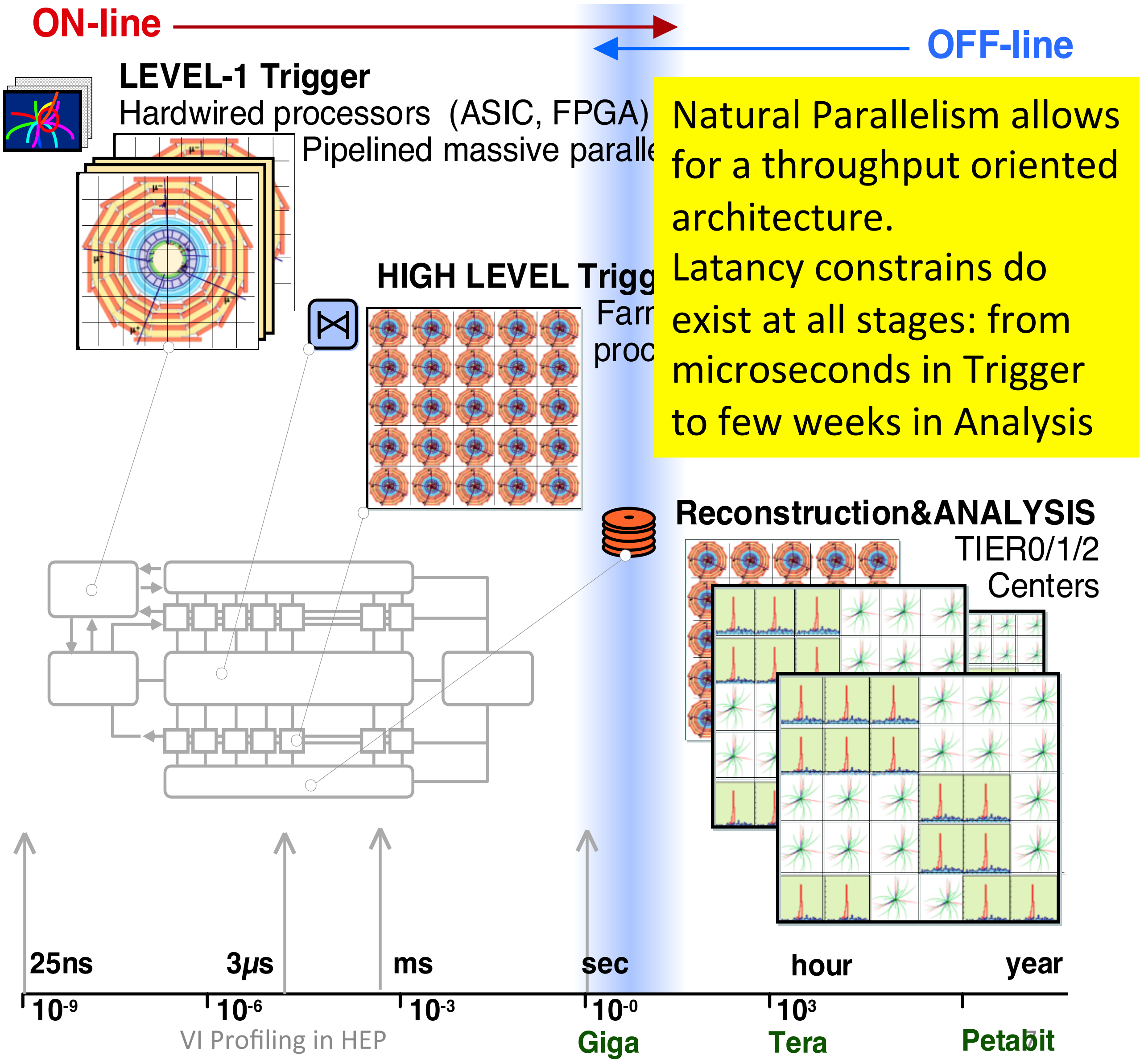
7 TeV Proton Proton
colliding beams

Bunch Crossing	**4 10^7 Hz**
Proton Collisions	**10^9 Hz**
Parton Collisions	
New Particle Production (Higgs, SUSY, ...)	**10^{-5} Hz**

Selection of 1 event in 10,000,000,000,000

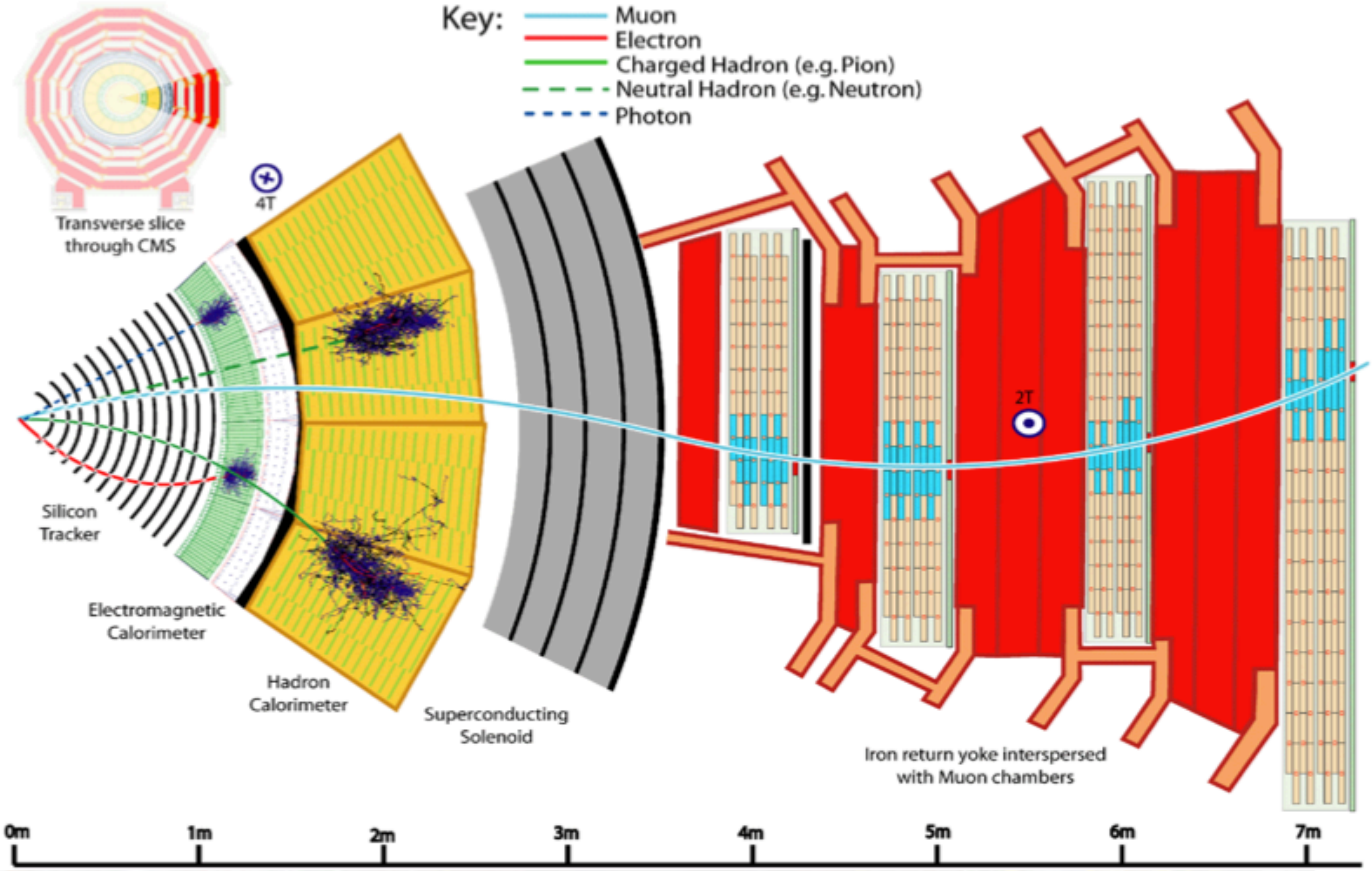
Data Flow



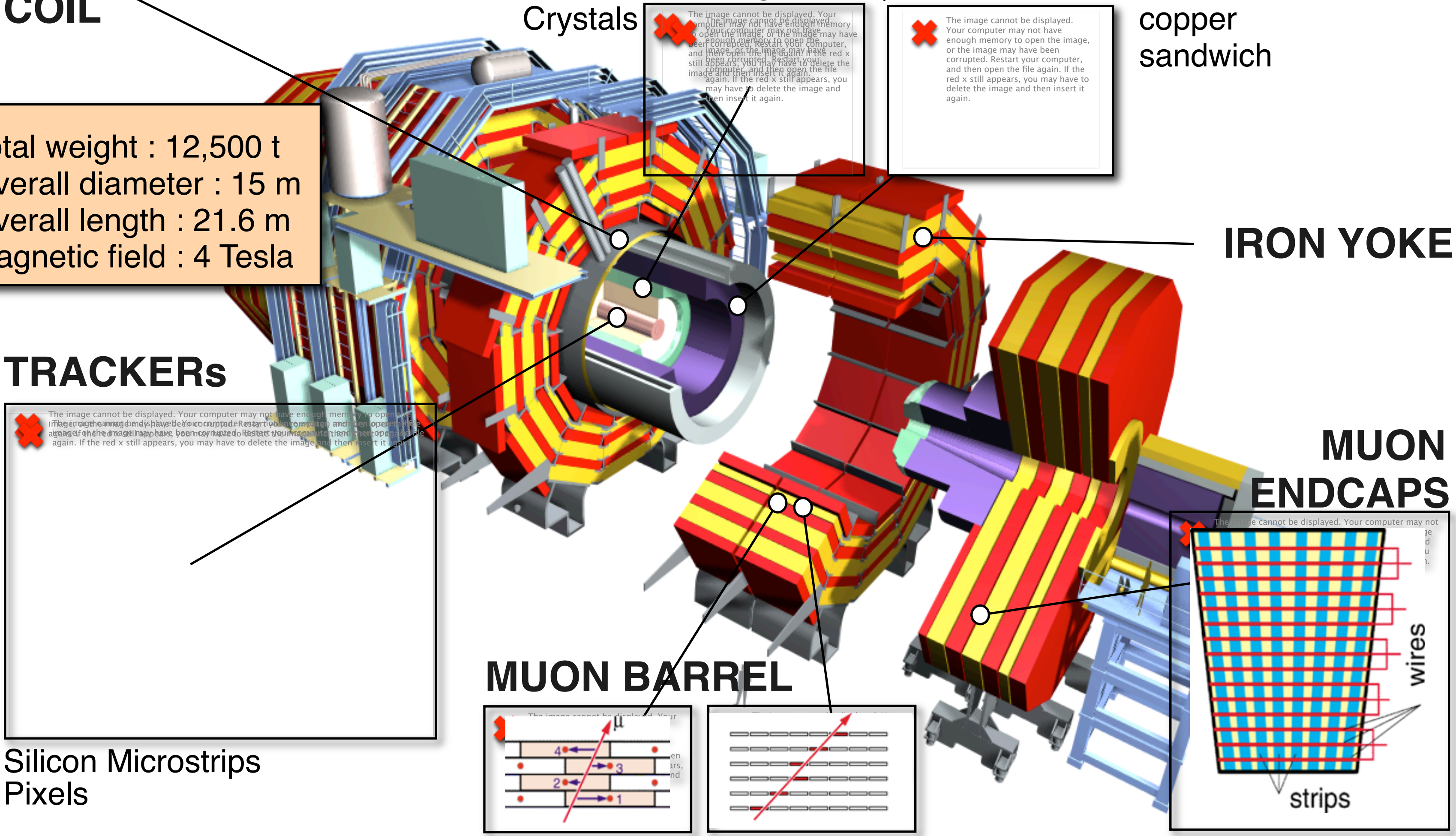


Natural Parallelism allows for a throughput oriented architecture.
Latancy constrains do exist at all stages: from microseconds in Trigger to few weeks in Analysis

Detector “onion” structure


Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
Transverse slice through CMS
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0m
1m
2m
3m
4m
5m
6m
7m

An experiment: CMS



SUPERCONDUCTING COIL

CALORIMETERS

ECAL Scintillating $PbWO_4$ Crystals

HCAL Plastic scintillator copper sandwich

Total weight : 12,500 t
Overall diameter : 15 m
Overall length : 21.6 m
Magnetic field : 4 Tesla

IRON YOKE

TRACKERs

Silicon Microstrips
Pixels

MUON ENDCAPS

MUON BARREL

Drift Tube Chambers (**DT**)

Resistive Plate Chambers (**RPC**)

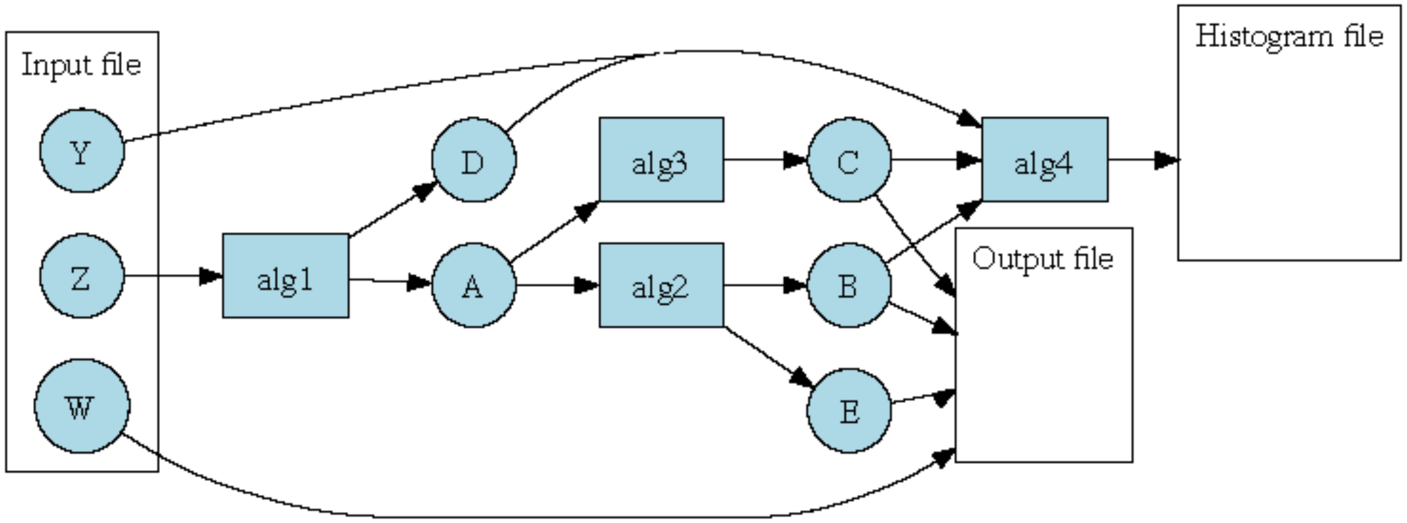
Cathode Strip Chambers (**CSC**)
Resistive Plate Chambers (**RPC**)

Data and Algorithms

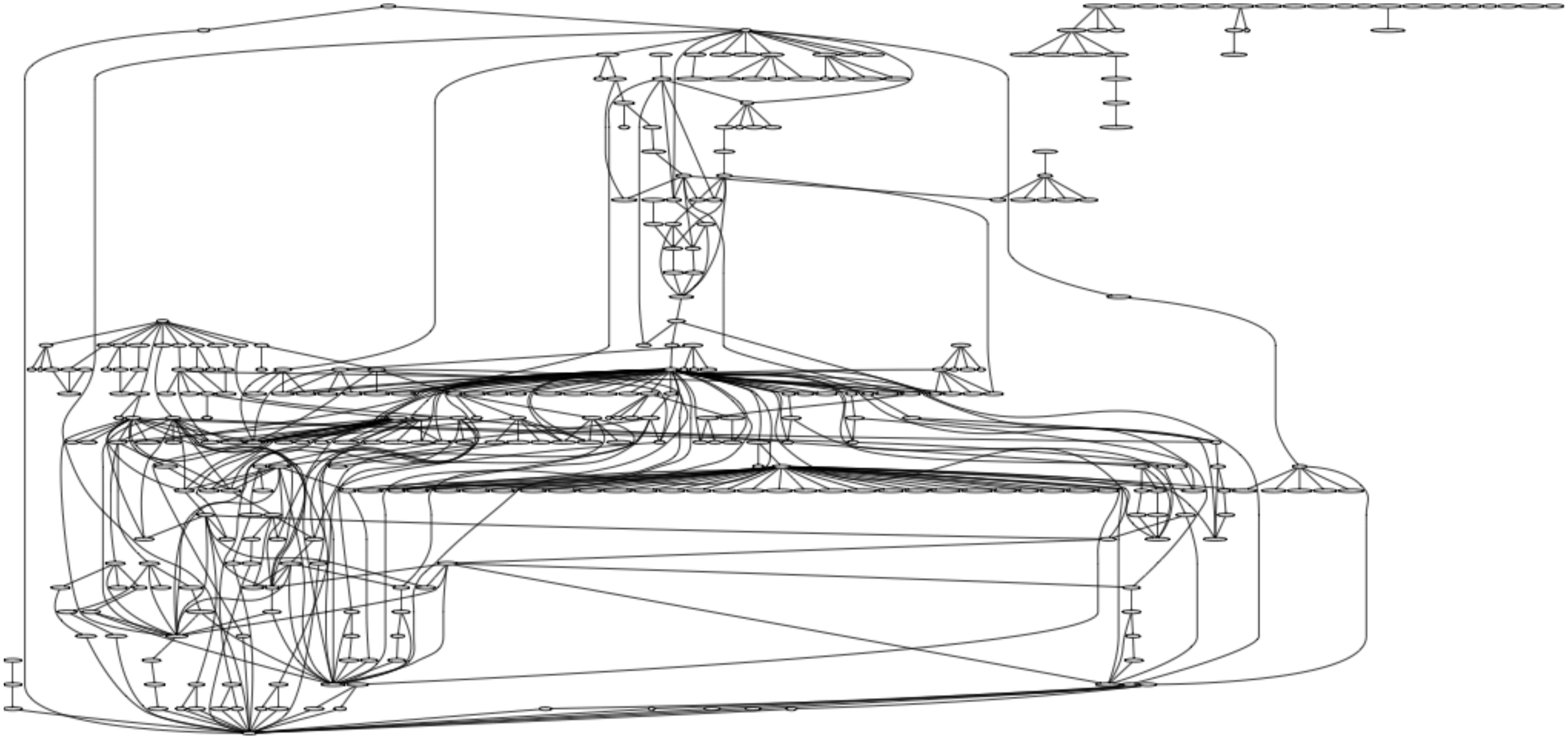
- HEP main data are organized in *Events* (particle collisions)
- Simulation, Reconstruction and Analysis programs process "one Event at the time"
  - Events are fairly independent of each other
  - "Natural" parallel processing
- Event processing programs are composed of a number of Algorithms selecting and transforming "raw" Event data into "processed" (reconstructed) Event data and statistics
  - Algorithms are mainly developed by "Physicists"
  - Algorithms may require additional "detector conditions" data (e.g. calibrations, geometry, environmental parameters, etc.)
  - Statistical data (histograms, distributions, etc.) are typically the final data processing results

A simplified data-processing application

- Circles represent data products
- Rectangles represent processing modules
- Any or all data products may be written to an output file

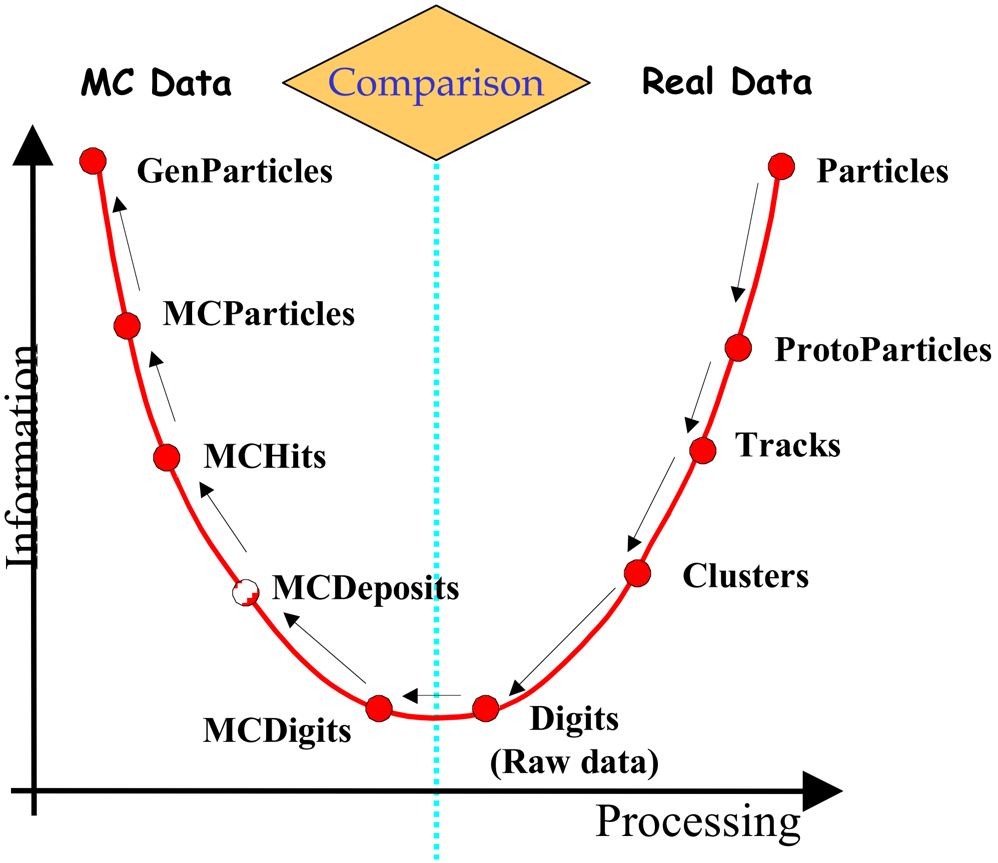


A real application (LHCb Brunel)



High Energy Analysis Model

MonteCarlo Simulation follows the evolution of physics processes from collision to digital signals



Reconstruction “goes back in time” from digital signals to the original particles produced in the collision

Analysis compares (at statistical level) reconstructed events from real data with those from simulation

19/11/13

now also for the general public:
http://cms.web.cern.ch/content/cms-physics-masterclass

Analogies with Industry

- Signal/image processing
  - Digital-Analog Conversions (including calibrations)
  - Pattern recognition, “clustering”
- Topological problems
  - Nearest neighbor, minimum path, space partitioning
- Gaming *(our main source of inspiration!)*
  - “walk-through” complex 3D geometries
  - Detection of “collisions”
- Navigation/Avionics (Kalman filtering)
  - Tracking in a force field in presence of “noise”
  - Trajectory identification and prediction

CMS simulation & data processing Software

- ~10k "modules"
- ~1000 "data processing" modules
- Code (SLOC)
  - C++: 3,558,032 (68.86%)
  - python: 1,240,801 (24.02%)
    - Used only in initialization
  - fortran: 277,857 (5.38%)
    - Interface to physics simulation code
- Total size of TEXT sections : 229,246,680 bytes
  - + ~220MB of "external software"

Main Data Processing Application

- 550 Module instances
  - Some share same code, different parameters or inputs
- ~200 MB of code loaded (as shared libraries)
  - Only 122MB active:
    - of which 50MB in "dictionaries" used in I/O
- Typical profile (by IgProf)
  - 8004 symbols visited (in 1,704,336 samples)
    - One sample each 0.006 seconds
  - 36495 total symbols recorded in the stack traces
  - Report was a 9MB sql3 db

Typical Profile (by “perfmon2”)

CPI (cycle per instruction): 0.964

load instructions %: 30.58%
store instructions %: 13.74%
branch instructions % (approx): 17.06%
resource stalls % (of cycles): 30.63%
divider busy % (of cycles): **12.11**%
% of branch instr. mispredicted: 2.25%
% of L3 loads missed: 2.09%

% of SIMD in all uops: 19.22%
% of comp. SIMD in all uops: 10.17%

breakdown:	%of all uops	% of all SIMD
PACKED_DOUBLE:	0.663%	3.449%
PACKED_SINGLE:	0.613%	3.190%
SCALAR_DOUBLE:	13.485%	**70.159**%
SCALAR_SINGLE:	4.038%	21.010%
VECTOR_INTEGER:	0.421%	2.192%

More details (see next page):

Function where time is spent most

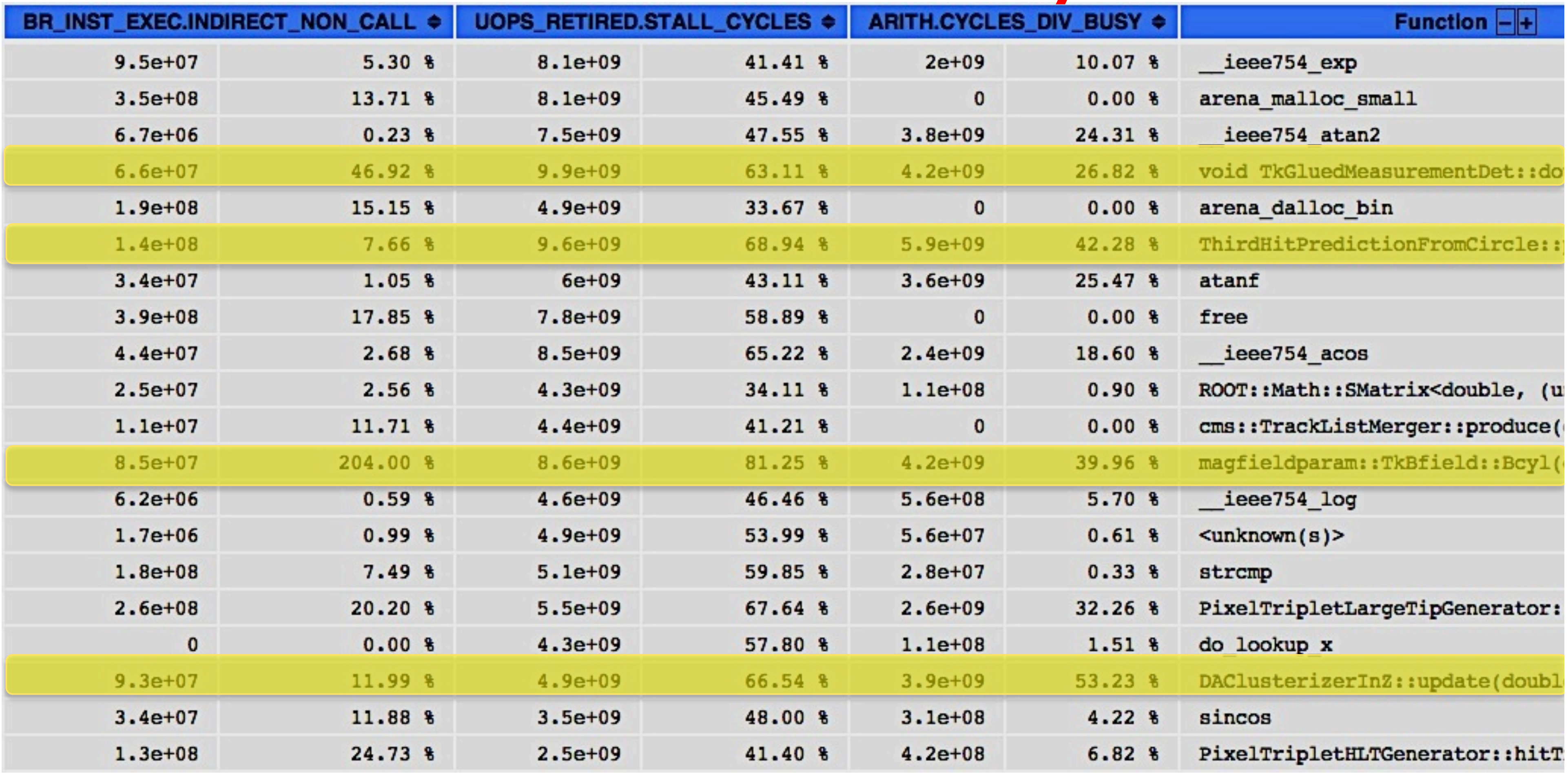
- *No hot-spots: top 30 each between 2.5% and 0.5% of total*
- Trig/trans functions, malloc…
- div/sqrt latency (*“divider busy”*)

“top self cost” (based on old PTU)

function calls latency

Stall time

div/sqrt latency



BR_INST_EXEC.INDIRECT_NON_CALL ◆		UOPS_RETIRED.STALL_CYCLES ◆		ARITH.CYCLES_DIV_BUSY ◆		Function
9.5e+07	5.30 %	8.1e+09	41.41 %	2e+09	10.07 %	__ieee754_exp
3.5e+08	13.71 %	8.1e+09	45.49 %	0	0.00 %	arena_malloc_small
6.7e+06	0.23 %	7.5e+09	47.55 %	3.8e+09	24.31 %	__ieee754_atan2
6.6e+07	46.92 %	9.9e+09	63.11 %	4.2e+09	26.82 %	void TkGluedMeasurementDet::do
1.9e+08	15.15 %	4.9e+09	33.67 %	0	0.00 %	arena_dalloc_bin
1.4e+08	7.66 %	9.6e+09	68.94 %	5.9e+09	42.28 %	ThirdHitPredictionFromCircle::
3.4e+07	1.05 %	6e+09	43.11 %	3.6e+09	25.47 %	atanf
3.9e+08	17.85 %	7.8e+09	58.89 %	0	0.00 %	free
4.4e+07	2.68 %	8.5e+09	65.22 %	2.4e+09	18.60 %	__ieee754_acos
2.5e+07	2.56 %	4.3e+09	34.11 %	1.1e+08	0.90 %	ROOT::Math::SMatrix<double, (u
1.1e+07	11.71 %	4.4e+09	41.21 %	0	0.00 %	cms::TrackListMerger::produce(
8.5e+07	204.00 %	8.6e+09	81.25 %	4.2e+09	39.96 %	magfieldparam::TkBfield::Bcyl(
6.2e+06	0.59 %	4.6e+09	46.46 %	5.6e+08	5.70 %	__ieee754_log
1.7e+06	0.99 %	4.9e+09	53.99 %	5.6e+07	0.61 %	<unknown(s)>
1.8e+08	7.49 %	5.1e+09	59.85 %	2.8e+07	0.33 %	strcmp
2.6e+08	20.20 %	5.5e+09	67.64 %	2.6e+09	32.26 %	PixelTripletLargeTipGenerator:
0	0.00 %	4.3e+09	57.80 %	1.1e+08	1.51 %	do_lookup_x
9.3e+07	11.99 %	4.9e+09	66.54 %	3.9e+09	53.23 %	DAClusterizerInZ::update(doubl
3.4e+07	11.88 %	3.5e+09	48.00 %	3.1e+08	4.22 %	sincos
1.3e+08	24.73 %	2.5e+09	41.40 %	4.2e+08	6.82 %	PixelTripletHLTGenerator::hitT

https://twiki.cern.ch/twiki/bin/view/LCG/VICMSDPHistory#Performance_Improvements
https://twiki.cern.ch/twiki/bin/view/LCG/MultiCoreRD#Track_1

PERFORMANCE MEASUREMENTS AND IMPROVEMENTS

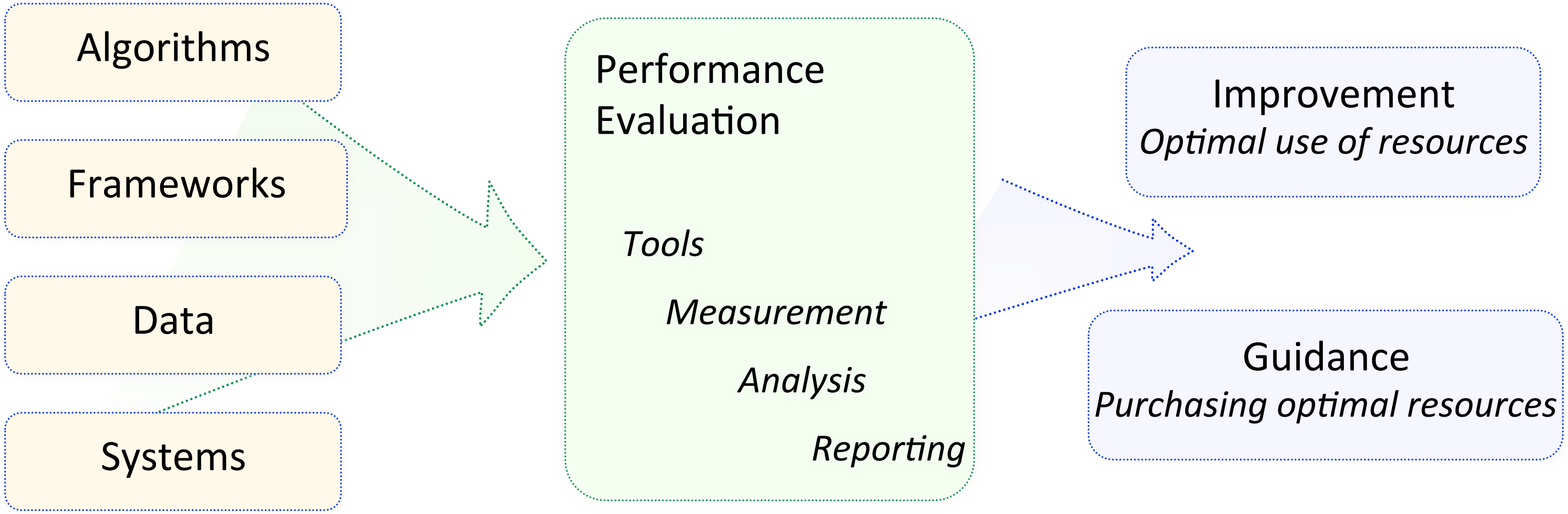
Performance evaluation/improvement Organization

Core team of specialists

- Evaluate the cost of running CMS software
- Develop, deploy tools
- Identify actions to reduce such a cost

Involvement of few scientist (students) with excellent software skills

- Algorithm modifications



Tool History (in CMS)

- OProfile (~2005)
- Callgrind
  - Added ability to resolve ancestors in "diamonds"
  - First attempt to summarize result in DB
- Development of in-house tool: igProf
- Perfmon
  - Instrumented Application with counting and sampling (leverage igprof technology)
  - Web reporting
- PTU
  - Added Web reporting
- VTune/Amplifier (just evaluation)
- Perf
  - Instrumented Application with counting

igprof

- CMS in-house performance and memory profiler (http://igprof.org).
  - *Designed to handle extremely large memory footprints and churn, which are typical of the HEP applications*
- Logically subdivided in three parts:
  - *dynamic instrumentation and code injection library (IgHook, inspired by work by Jeffrey Richter, Jonathan Rentzsch, Secure Reality, and of course Sun DTrace).*
  - *the profiler itself, which instruments various system calls and keeps track of SIG_PROFILE occurrences (for sampled application profiling) and memory allocation operations (for various kind of memory profiling)*
  - *the analyzer and the display tool, which read the output of the profiler and provide a textual or web based view of the information.*
- Supports multiple platforms, mainly x86 but has support for PPC and now initial ARM support. Heavily dependent on libunwind.

igprof

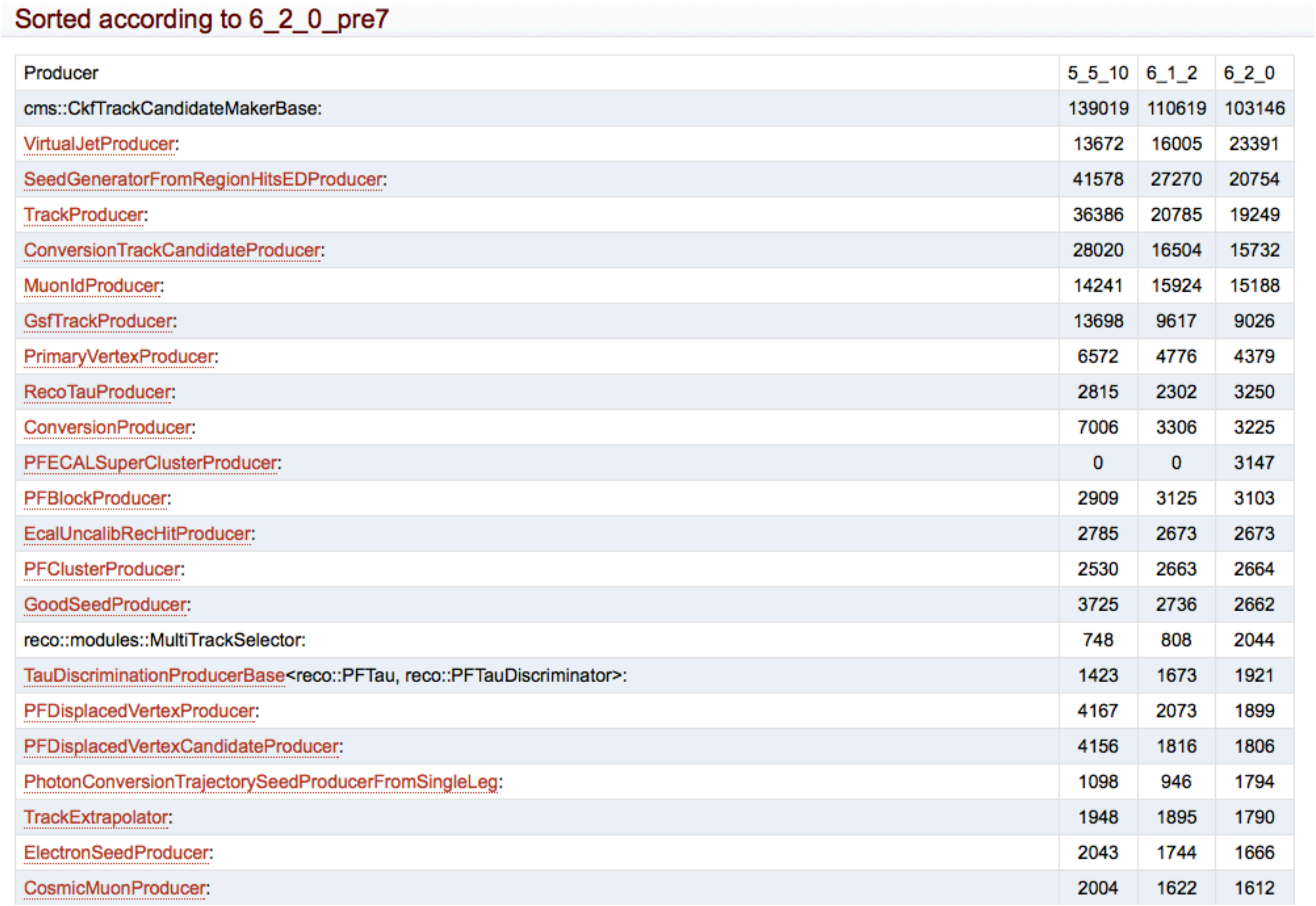
- What igprof can tell you:
  - *How much time (statistically) you spend in a given function and which fraction of the cost occurs when called by / calling someone else.*
  - *Tracks all live memory allocations at the moment of the dump, providing memory map, aggregate (per call-stack) statistics about total size and size of the largest allocation in a given call-stack. Tracks sources of allocations.*
  - *Instruments and precisely measure time spent in a given function(s).*

THE TOOL: General Requirements

- Run in user space
- Under Linux (and Mac-OS)
- In standard production environment
- No special compilation
- Ability to instrument framework and user code at user specified granularity
- Generate results viewable anywhere, anytime

Requirements: 1

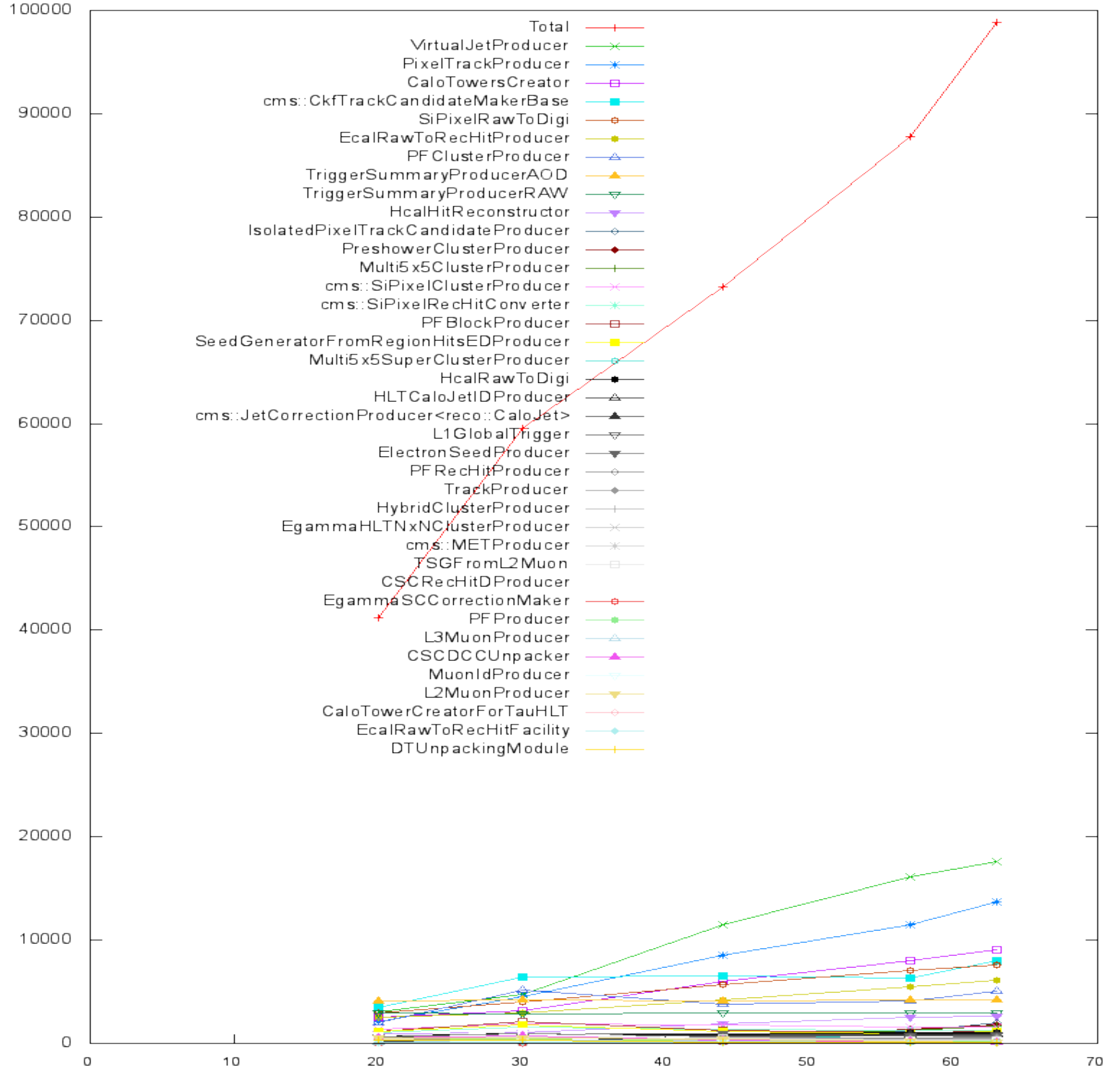
- “Manager” view
  - Low granularity (often application dependent)
  - Coverage: full workflow
  - Regression among releases
  - Trend as function of “external conditions”
  - Spot offenders
  - Easy to publish, consult, archive
  - Deliverable: **assessment reports**



Sorted according to 6_2_0_pre7

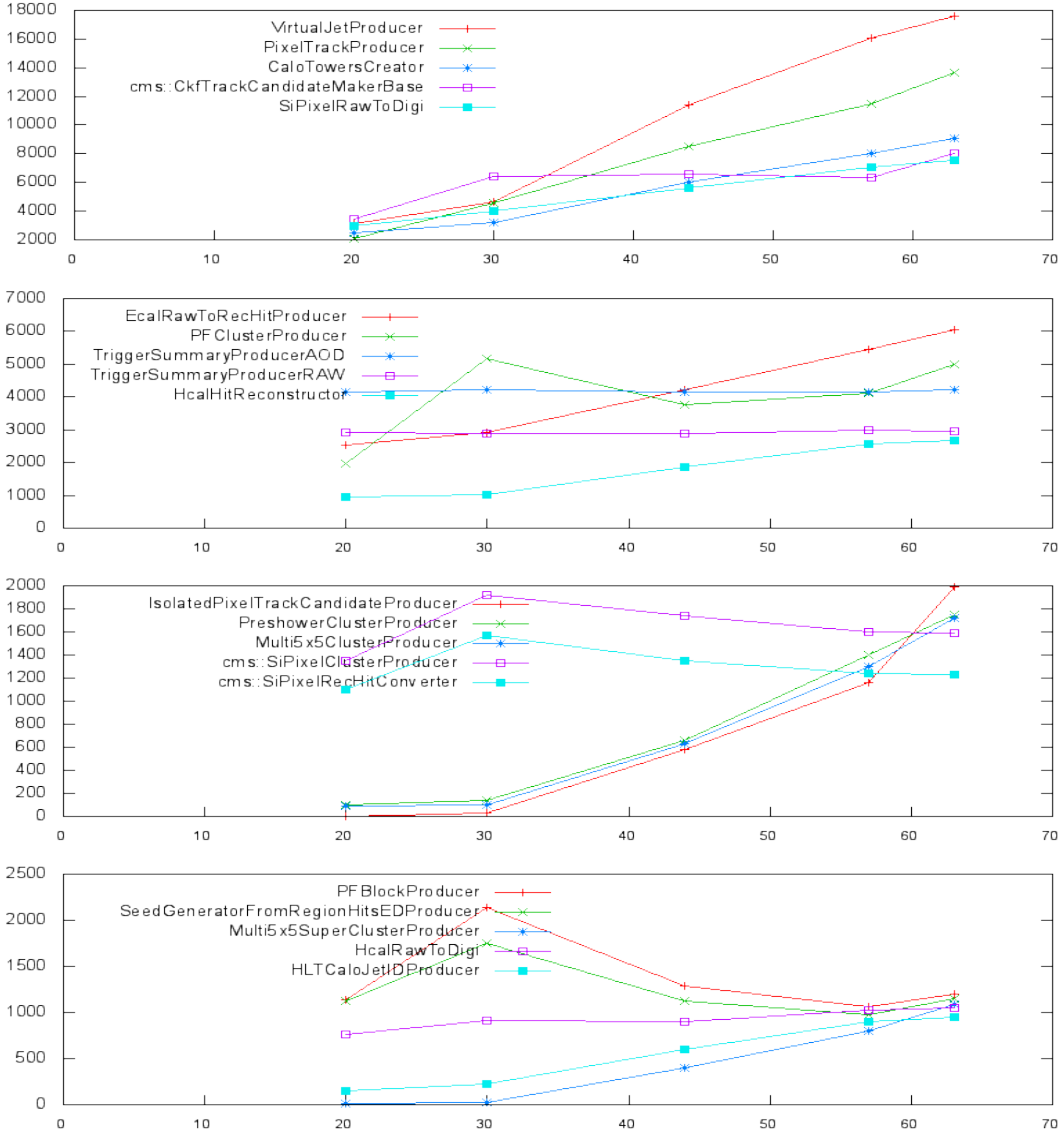
Producer	5_5_10	6_1_2	6_2_0
cms::CkfTrackCandidateMakerBase:	139019	110619	103146
VirtualJetProducer:	13672	16005	23391
SeedGeneratorFromRegionHitsEDProducer:	41578	27270	20754
TrackProducer:	36386	20785	19249
ConversionTrackCandidateProducer:	28020	16504	15732
MuonIdProducer:	14241	15924	15188
GsfTrackProducer:	13698	9617	9026
PrimaryVertexProducer:	6572	4776	4379
RecoTauProducer:	2815	2302	3250
ConversionProducer:	7006	3306	3225
PFECALSuperClusterProducer:	0	0	3147
PFBlockProducer:	2909	3125	3103
EcalUncalibRecHitProducer:	2785	2673	2673
PFClusterProducer:	2530	2663	2664
GoodSeedProducer:	3725	2736	2662
reco::modules::MultiTrackSelector:	748	808	2044
TauDiscriminationProducerBase<reco::PFTau, reco::PFTauDiscriminator>:	1423	1673	1921
PFDisplacedVertexProducer:	4167	2073	1899
PFDisplacedVertexCandidateProducer:	4156	1816	1806
PhotonConversionTrajectorySeedProducerFromSingleLeg:	1098	946	1794
TrackExtrapolator:	1948	1895	1790
ElectronSeedProducer:	2043	1744	1666
CosmicMuonProducer:	2004	1622	1612

Time in online Filter vs #collisions



Total is the sum of many "small" contributions.
ALL need to be optimized to produce an observable HLT speed-up

Top 20 contributors

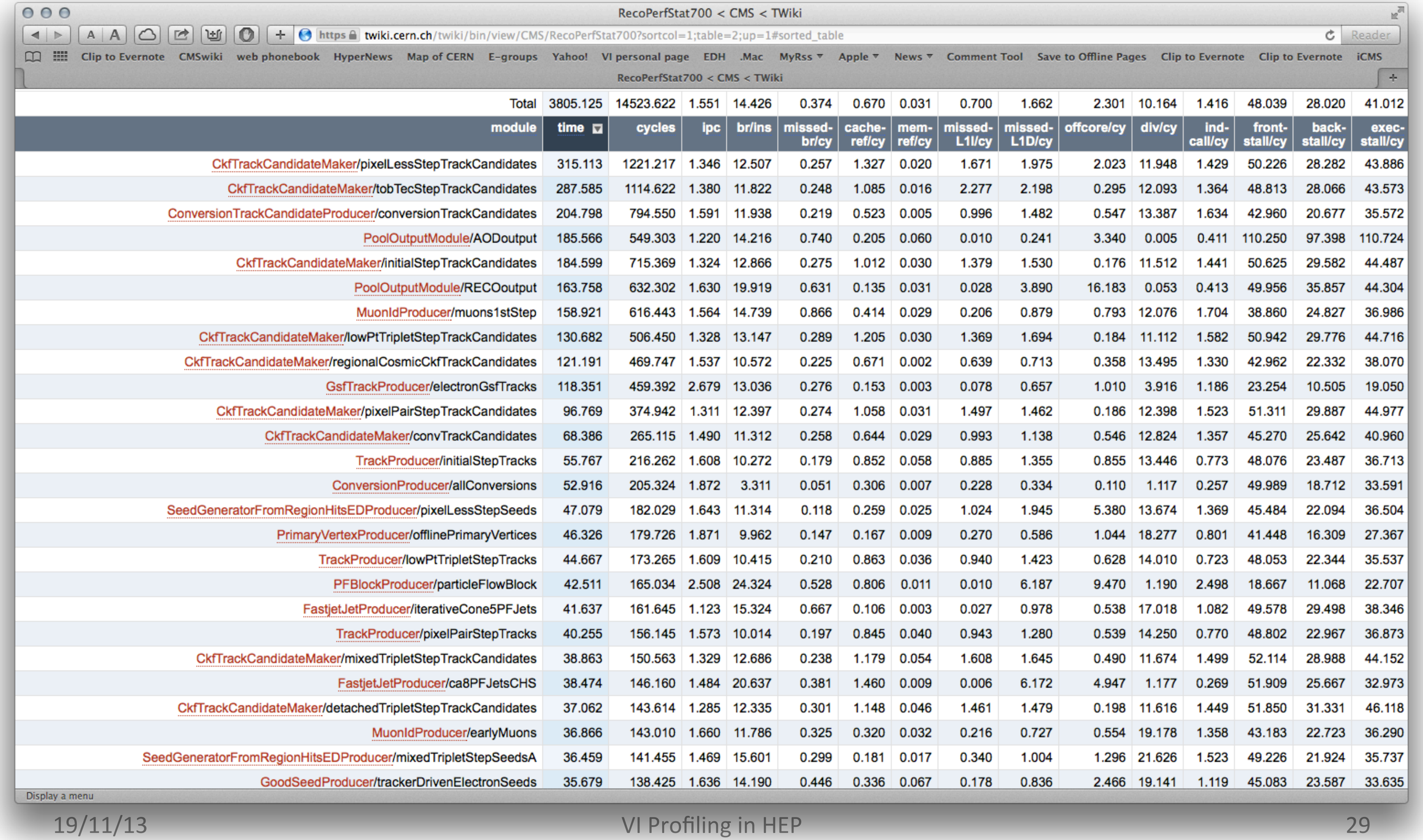


Large: still almost Linear scaling

Tracking: affected by changes in menu in HighPU run

Potentially small: Scary exponential scaling

Module level instrumentation with “perflib”

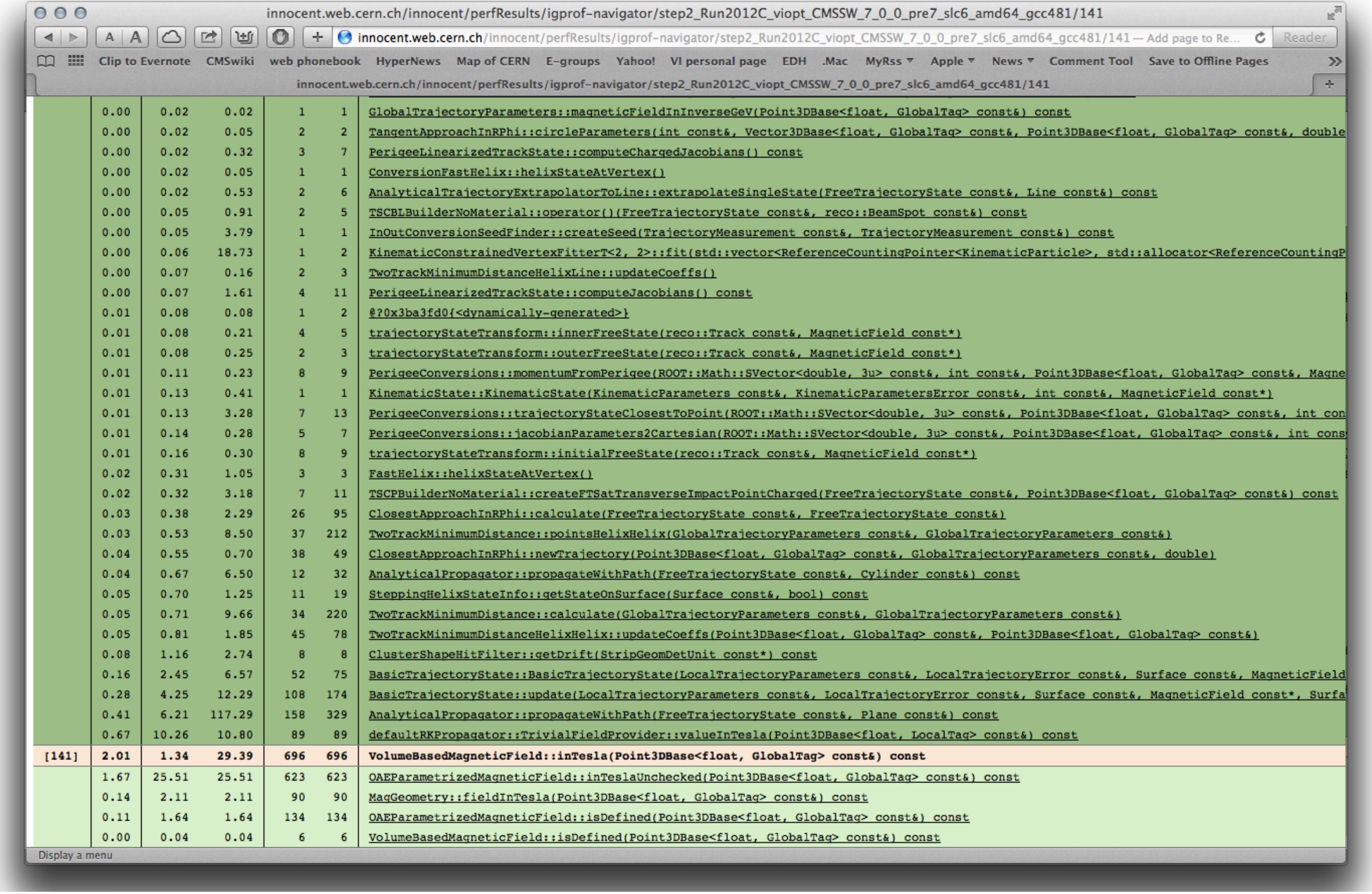

RecoPerfStat700 < CMS < TWiki
https twiki.cern.ch/twiki/bin/view/CMS/RecoPerfStat700?sortcol=1;table=2;up=1#sorted_table
Reader
Clip to Evernote CMSwiki web phonebook HyperNews Map of CERN E-groups Yahoo! VI personal page EDH .Mac MyRss Apple News Comment Tool Save to Offline Pages Clip to Evernote Clip to Evernote iCMS
RecoPerfStat700 < CMS < TWiki
Display a menu

module	time	cycles	ipc	br/ins	missed-br/cy	cache-ref/cy	mem-ref/cy	missed-L1I/cy	missed-L1D/cy	offcore/cy	div/cy	ind-call/cy	front-stall/cy	back-stall/cy	exec-stall/cy
Total	3805.125	14523.622	1.551	14.426	0.374	0.670	0.031	0.700	1.662	2.301	10.164	1.416	48.039	28.020	41.012
CkfTrackCandidateMaker/pixelLessStepTrackCandidates	315.113	1221.217	1.346	12.507	0.257	1.327	0.020	1.671	1.975	2.023	11.948	1.429	50.226	28.282	43.886
CkfTrackCandidateMaker/tobTecStepTrackCandidates	287.585	1114.622	1.380	11.822	0.248	1.085	0.016	2.277	2.198	0.295	12.093	1.364	48.813	28.066	43.573
ConversionTrackCandidateProducer/conversionTrackCandidates	204.798	794.550	1.591	11.938	0.219	0.523	0.005	0.996	1.482	0.547	13.387	1.634	42.960	20.677	35.572
PoolOutputModule/AODoutput	185.566	549.303	1.220	14.216	0.740	0.205	0.060	0.010	0.241	3.340	0.005	0.411	110.250	97.398	110.724
CkfTrackCandidateMaker/initialStepTrackCandidates	184.599	715.369	1.324	12.866	0.275	1.012	0.030	1.379	1.530	0.176	11.512	1.441	50.625	29.582	44.487
PoolOutputModule/RECOoutput	163.758	632.302	1.630	19.919	0.631	0.135	0.031	0.028	3.890	16.183	0.053	0.413	49.956	35.857	44.304
MuonIdProducer/muons1stStep	158.921	616.443	1.564	14.739	0.866	0.414	0.029	0.206	0.879	0.793	12.076	1.704	38.860	24.827	36.986
CkfTrackCandidateMaker/lowPtTripletStepTrackCandidates	130.682	506.450	1.328	13.147	0.289	1.205	0.030	1.369	1.694	0.184	11.112	1.582	50.942	29.776	44.716
CkfTrackCandidateMaker/regionalCosmicCkfTrackCandidates	121.191	469.747	1.537	10.572	0.225	0.671	0.002	0.639	0.713	0.358	13.495	1.330	42.962	22.332	38.070
GsfTrackProducer/electronGsfTracks	118.351	459.392	2.679	13.036	0.276	0.153	0.003	0.078	0.657	1.010	3.916	1.186	23.254	10.505	19.050
CkfTrackCandidateMaker/pixelPairStepTrackCandidates	96.769	374.942	1.311	12.397	0.274	1.058	0.031	1.497	1.462	0.186	12.398	1.523	51.311	29.887	44.977
CkfTrackCandidateMaker/convTrackCandidates	68.386	265.115	1.490	11.312	0.258	0.644	0.029	0.993	1.138	0.546	12.824	1.357	45.270	25.642	40.960
TrackProducer/initialStepTracks	55.767	216.262	1.608	10.272	0.179	0.852	0.058	0.885	1.355	0.855	13.446	0.773	48.076	23.487	36.713
ConversionProducer/allConversions	52.916	205.324	1.872	3.311	0.051	0.306	0.007	0.228	0.334	0.110	1.117	0.257	49.989	18.712	33.591
SeedGeneratorFromRegionHitsEDProducer/pixelLessStepSeeds	47.079	182.029	1.643	11.314	0.118	0.259	0.025	1.024	1.945	5.380	13.674	1.369	45.484	22.094	36.504
PrimaryVertexProducer/offlinePrimaryVertices	46.326	179.726	1.871	9.962	0.147	0.167	0.009	0.270	0.586	1.044	18.277	0.801	41.448	16.309	27.367
TrackProducer/lowPtTripletStepTracks	44.667	173.265	1.609	10.415	0.210	0.863	0.036	0.940	1.423	0.628	14.010	0.723	48.053	22.344	35.537
PFBlockProducer/particleFlowBlock	42.511	165.034	2.508	24.324	0.528	0.806	0.011	0.010	6.187	9.470	1.190	2.498	18.667	11.068	22.707
FastjetJetProducer/iterativeCone5PFJets	41.637	161.645	1.123	15.324	0.667	0.106	0.003	0.027	0.978	0.538	17.018	1.082	49.578	29.498	38.346
TrackProducer/pixelPairStepTracks	40.255	156.145	1.573	10.014	0.197	0.845	0.040	0.943	1.280	0.539	14.250	0.770	48.802	22.967	36.873
CkfTrackCandidateMaker/mixedTripletStepTrackCandidates	38.863	150.563	1.329	12.686	0.238	1.179	0.054	1.608	1.645	0.490	11.674	1.499	52.114	28.988	44.152
FastjetJetProducer/ca8PFJetsCHS	38.474	146.160	1.484	20.637	0.381	1.460	0.009	0.006	6.172	4.947	1.177	0.269	51.909	25.667	32.973
CkfTrackCandidateMaker/detachedTripletStepTrackCandidates	37.062	143.614	1.285	12.335	0.301	1.148	0.046	1.461	1.479	0.198	11.616	1.449	51.850	31.331	46.118
MuonIdProducer/earlyMuons	36.866	143.010	1.660	11.786	0.325	0.320	0.032	0.216	0.727	0.554	19.178	1.358	43.183	22.723	36.290
SeedGeneratorFromRegionHitsEDProducer/mixedTripletStepSeedsA	36.459	141.455	1.469	15.601	0.299	0.181	0.017	0.340	1.004	1.296	21.626	1.523	49.226	21.924	35.737
GoodSeedProducer/trackerDrivenElectronSeeds	35.679	138.425	1.636	14.190	0.446	0.336	0.067	0.178	0.836	2.466	19.141	1.119	45.083	23.587	33.635

Requirements: 2

- "Efficiency Officer" View
  - Function level granularity (with stack trace!)
  - Coverage: Application
  - Identification of hot-spot
  - Correlation with code (and data?)
  - Guidance for optimization opportunities
  - Easy to publish, consult, archive, trace improvements
  - Deliverable: **detailed bug reports**

Igprof callers and callees

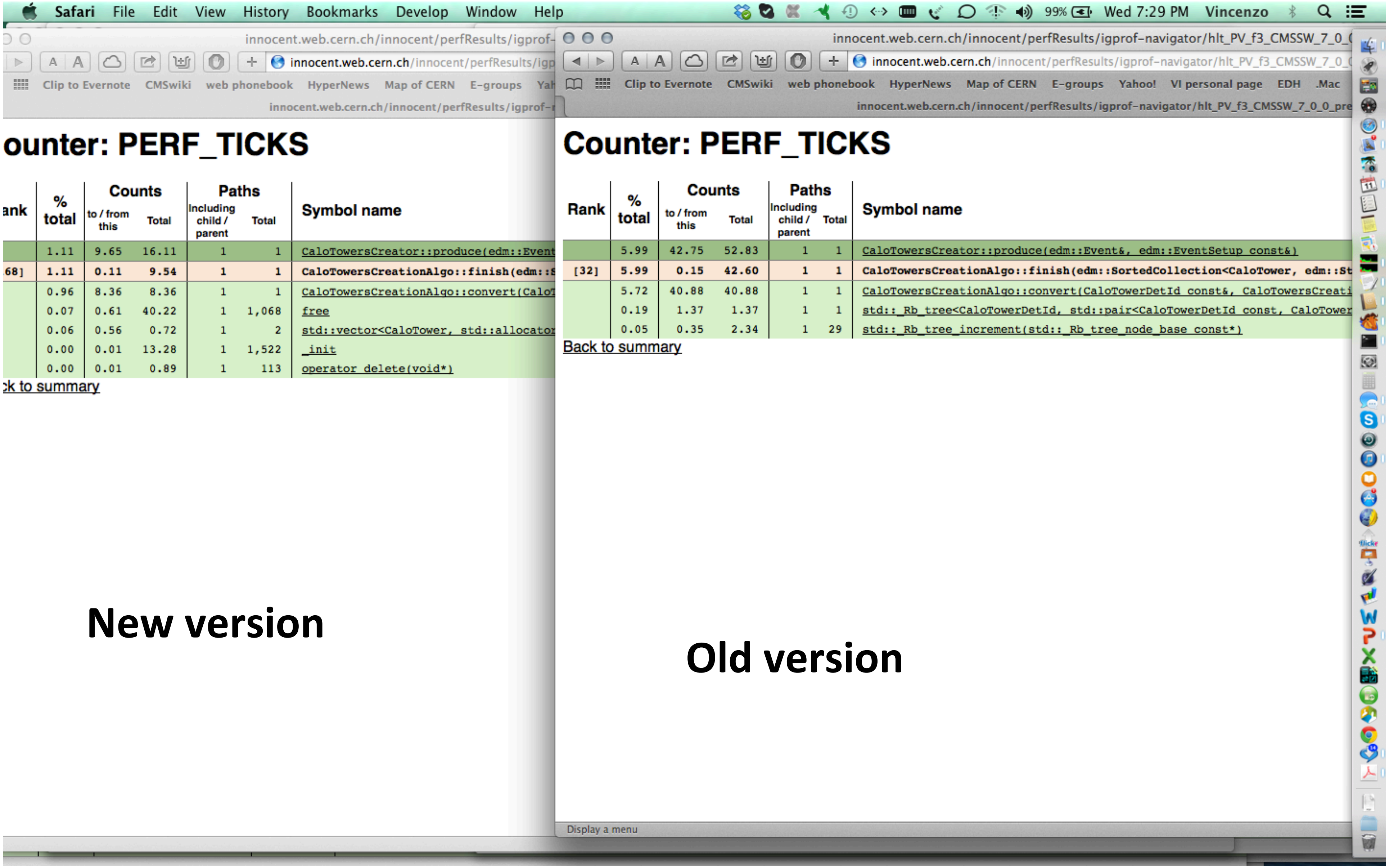

innocent.web.cern.ch/innocent/perfResults/igprof-navigator/step2_Run2012C_viopt_CMSSW_7_0_0_pre7_slc6_amd64_gcc481/141
innocent.web.cern.ch/innocent/perfResults/igprof-navigator/step2_Run2012C_viopt_CMSSW_7_0_0_pre7_slc6_amd64_gcc481/141 — Add page to Re...
Reader
Clip to Evernote
CMSwiki
web phonebook
HyperNews
Map of CERN
E-groups
Yahoo!
VI personal page
EDH
.Mac
MyRss
Apple
News
Comment Tool
Save to Offline Pages
innocent.web.cern.ch/innocent/perfResults/igprof-navigator/step2_Run2012C_viopt_CMSSW_7_0_0_pre7_slc6_amd64_gcc481/141
Display a menu

	0.00	0.02	0.02	1	1	GlobalTrajectoryParameters::magneticFieldInInverseGeV(Point3DBase<float, GlobalTag> const&) const
	0.00	0.02	0.05	2	2	TangentApproachInRPhi::circleParameters(int const&, Vector3DBase<float, GlobalTag> const&, Point3DBase<float, GlobalTag> const&, double
	0.00	0.02	0.32	3	7	PerigeeLinearizedTrackState::computeChargedJacobians() const
	0.00	0.02	0.05	1	1	ConversionFastHelix::helixStateAtVertex()
	0.00	0.02	0.53	2	6	AnalyticalTrajectoryExtrapolatorToLine::extrapolateSingleState(FreeTrajectoryState const&, Line const&) const
	0.00	0.05	0.91	2	5	TSCBLBuilderNoMaterial::operator()(FreeTrajectoryState const&, reco::BeamSpot const&) const
	0.00	0.05	3.79	1	1	InOutConversionSeedFinder::createSeed(TrajectoryMeasurement const&, TrajectoryMeasurement const&) const
	0.00	0.06	18.73	1	2	KinematicConstrainedVertexFitterT<2, 2>::fit(std::vector<ReferenceCountingPointer<KinematicParticle>, std::allocator<ReferenceCountingP
	0.00	0.07	0.16	2	3	TwoTrackMinimumDistanceHelixLine::updateCoeffs()
	0.00	0.07	1.61	4	11	PerigeeLinearizedTrackState::computeJacobians() const
	0.01	0.08	0.08	1	2	@?0x3ba3fd0{<dynamically-generated>}
	0.01	0.08	0.21	4	5	trajectoryStateTransform::innerFreeState(reco::Track const&, MagneticField const*)
	0.01	0.08	0.25	2	3	trajectoryStateTransform::outerFreeState(reco::Track const&, MagneticField const*)
	0.01	0.11	0.23	8	9	PerigeeConversions::momentumFromPerigee(ROOT::Math::SVector<double, 3u> const&, int const&, Point3DBase<float, GlobalTag> const&, Magne
	0.01	0.13	0.41	1	1	KinematicState::KinematicState(KinematicParameters const&, KinematicParametersError const&, int const&, MagneticField const*)
	0.01	0.13	3.28	7	13	PerigeeConversions::trajectoryStateClosestToPoint(ROOT::Math::SVector<double, 3u> const&, Point3DBase<float, GlobalTag> const&, int con
	0.01	0.14	0.28	5	7	PerigeeConversions::jacobianParameters2Cartesian(ROOT::Math::SVector<double, 3u> const&, Point3DBase<float, GlobalTag> const&, int cons
	0.01	0.16	0.30	8	9	trajectoryStateTransform::initialFreeState(reco::Track const&, MagneticField const*)
	0.02	0.31	1.05	3	3	FastHelix::helixStateAtVertex()
	0.02	0.32	3.18	7	11	TSCPBuilderNoMaterial::createFTSatTransverseImpactPointCharged(FreeTrajectoryState const&, Point3DBase<float, GlobalTag> const&) const
	0.03	0.38	2.29	26	95	ClosestApproachInRPhi::calculate(FreeTrajectoryState const&, FreeTrajectoryState const&)
	0.03	0.53	8.50	37	212	TwoTrackMinimumDistance::pointsHelixHelix(GlobalTrajectoryParameters const&, GlobalTrajectoryParameters const&)
	0.04	0.55	0.70	38	49	ClosestApproachInRPhi::newTrajectory(Point3DBase<float, GlobalTag> const&, GlobalTrajectoryParameters const&, double)
	0.04	0.67	6.50	12	32	AnalyticalPropagator::propagateWithPath(FreeTrajectoryState const&, Cylinder const&) const
	0.05	0.70	1.25	11	19	SteppingHelixStateInfo::getStateOnSurface(Surface const&, bool) const
	0.05	0.71	9.66	34	220	TwoTrackMinimumDistance::calculate(GlobalTrajectoryParameters const&, GlobalTrajectoryParameters const&)
	0.05	0.81	1.85	45	78	TwoTrackMinimumDistanceHelixHelix::updateCoeffs(Point3DBase<float, GlobalTag> const&, Point3DBase<float, GlobalTag> const&)
	0.08	1.16	2.74	8	8	ClusterShapeHitFilter::getDrift(StripGeomDetUnit const*) const
	0.16	2.45	6.57	52	75	BasicTrajectoryState::BasicTrajectoryState(LocalTrajectoryParameters const&, LocalTrajectoryError const&, Surface const&, MagneticField
	0.28	4.25	12.29	108	174	BasicTrajectoryState::update(LocalTrajectoryParameters const&, LocalTrajectoryError const&, Surface const&, MagneticField const*, Surfa
	0.41	6.21	117.29	158	329	AnalyticalPropagator::propagateWithPath(FreeTrajectoryState const&, Plane const&) const
	0.67	10.26	10.80	89	89	defaultRKPropagator::TrivialFieldProvider::valueInTesla(Point3DBase<float, LocalTag> const&) const
[141]	**2.01**	**1.34**	**29.39**	**696**	**696**	**VolumeBasedMagneticField::inTesla(Point3DBase<float, GlobalTag> const&) const**
	1.67	25.51	25.51	623	623	OAEParametrizedMagneticField::inTeslaUnchecked(Point3DBase<float, GlobalTag> const&) const
	0.14	2.11	2.11	90	90	MagGeometry::fieldInTesla(Point3DBase<float, GlobalTag> const&) const
	0.11	1.64	1.64	134	134	OAEParametrizedMagneticField::isDefined(Point3DBase<float, GlobalTag> const&) const
	0.00	0.04	0.04	6	6	VolumeBasedMagneticField::isDefined(Point3DBase<float, GlobalTag> const&) const

Requirements: 3

- Software Developer View
  - Fast turn-around (zero overhead)
  - Highest granularity
    - readable also for non specialists
  - Correlation with code
  - Flexible reporting (local and global impact)
  - Trace of improvements
  - Ability to publish and archive results
  - Deliverable: **better code!**

Igprof web navigator



Safari File Edit View History Bookmarks Develop Window Help
99% Wed 7:29 PM Vincenzo
innocent.web.cern.ch/innocent/perfResults/igprof-
innocent.web.cern.ch/innocent/perfResults/igprof-navigator/hlt_PV_f3_CMSSW_7_0_
Clip to Evernote CMSwiki web phonebook HyperNews Map of CERN E-groups Yahoo! VI personal page EDH .Mac
innocent.web.cern.ch/innocent/perfResults/igprof-navigator/hlt_PV_f3_CMSSW_7_0_0_pre
Display a menu

ounter: PERF_TICKS

ank	% total	Counts		Paths		Symbol name
		to / from this	Total	Including child / parent	Total	
	1.11	9.65	16.11	1	1	CaloTowersCreator::produce(edm::Event
68]	1.11	0.11	9.54	1	1	CaloTowersCreationAlgo::finish(edm::S
	0.96	8.36	8.36	1	1	CaloTowersCreationAlgo::convert(CaloT
	0.07	0.61	40.22	1	1,068	free
	0.06	0.56	0.72	1	2	std::vector<CaloTower, std::allocator
	0.00	0.01	13.28	1	1,522	_init
	0.00	0.01	0.89	1	113	operator delete(void*)

ck to summary

Counter: PERF_TICKS

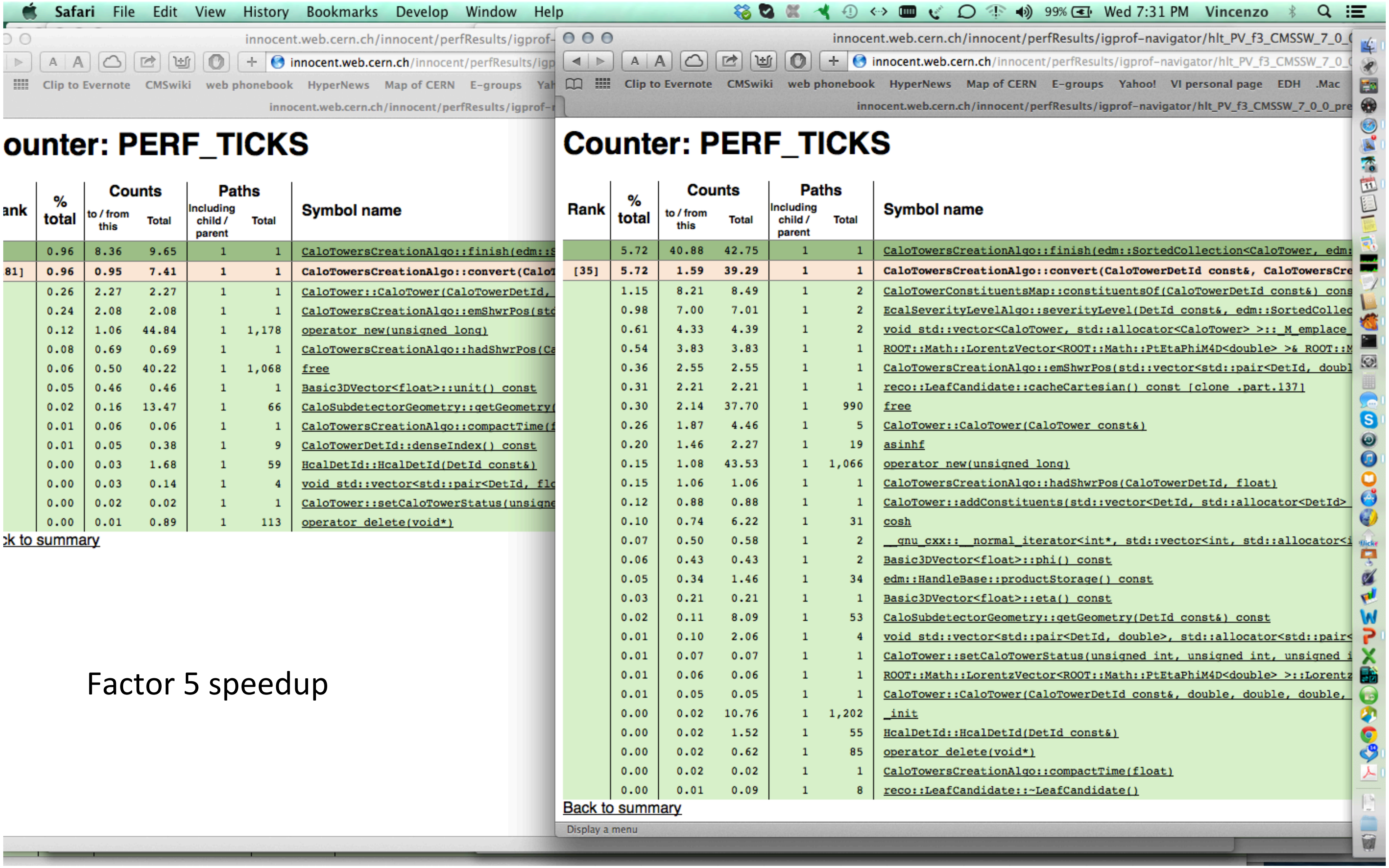
Rank	% total	Counts		Paths		Symbol name
		to / from this	Total	Including child / parent	Total	
	5.99	42.75	52.83	1	1	CaloTowersCreator::produce(edm::Event&, edm::EventSetup const&)
[32]	5.99	0.15	42.60	1	1	CaloTowersCreationAlgo::finish(edm::SortedCollection<CaloTower, edm::St
	5.72	40.88	40.88	1	1	CaloTowersCreationAlgo::convert(CaloTowerDetId const&, CaloTowersCreati
	0.19	1.37	1.37	1	1	std::_Rb_tree<CaloTowerDetId, std::pair<CaloTowerDetId const, CaloTower
	0.05	0.35	2.34	1	29	std::_Rb_tree_increment(std::_Rb_tree_node_base const*)

Back to summary

New version

Old version

Down into “convert”


Safari File Edit View History Bookmarks Develop Window Help
99% Wed 7:31 PM Vincenzo
innocent.web.cern.ch/innocent/perfResults/igprof-
innocent.web.cern.ch/innocent/perfResults/igprof-navigator/hlt_PV_f3_CMSSW_7_0_
Clip to Evernote CMSwiki web phonebook HyperNews Map of CERN E-groups Yahoo! VI personal page EDH .Mac
innocent.web.cern.ch/innocent/perfResults/igprof-navigator/hlt_PV_f3_CMSSW_7_0_0_pre
Back to summary
Display a menu

ounter: PERF_TICKS

ank	% total	Counts		Paths		Symbol name
		to / from this	Total	Including child / parent	Total	
	0.96	8.36	9.65	1	1	CaloTowersCreationAlgo::finish(edm::S
81]	0.96	0.95	7.41	1	1	**CaloTowersCreationAlgo::convert(CaloT**
	0.26	2.27	2.27	1	1	CaloTower::CaloTower(CaloTowerDetId,
	0.24	2.08	2.08	1	1	CaloTowersCreationAlgo::emShwrPos(std
	0.12	1.06	44.84	1	1,178	operator new(unsigned long)
	0.08	0.69	0.69	1	1	CaloTowersCreationAlgo::hadShwrPos(Ca
	0.06	0.50	40.22	1	1,068	free
	0.05	0.46	0.46	1	1	Basic3DVector<float>::unit() const
	0.02	0.16	13.47	1	66	CaloSubdetectorGeometry::getGeometry(
	0.01	0.06	0.06	1	1	CaloTowersCreationAlgo::compactTime(f
	0.01	0.05	0.38	1	9	CaloTowerDetId::denseIndex() const
	0.00	0.03	1.68	1	59	HcalDetId::HcalDetId(DetId const&)
	0.00	0.03	0.14	1	4	void std::vector<std::pair<DetId, flo
	0.00	0.02	0.02	1	1	CaloTower::setCaloTowerStatus(unsigne
	0.00	0.01	0.89	1	113	operator delete(void*)

ck to summary

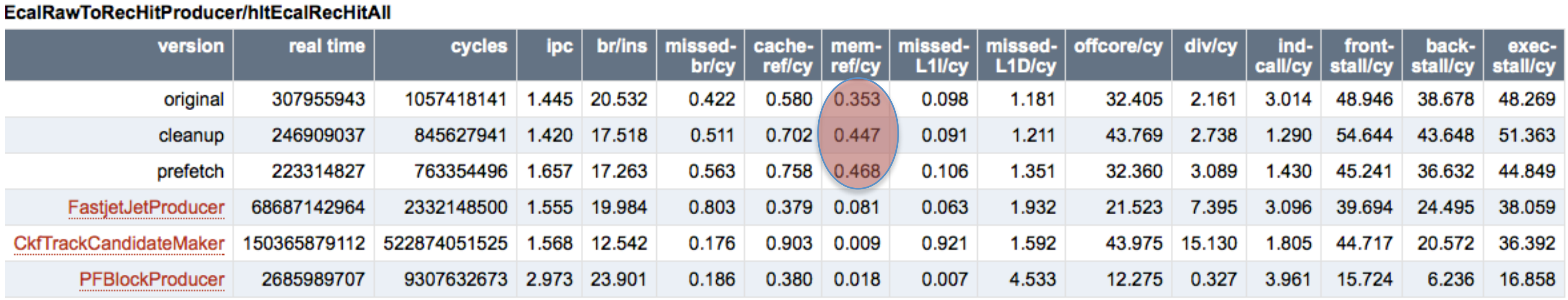
Factor 5 speedup

Counter: PERF_TICKS

Rank	% total	Counts		Paths		Symbol name
		to / from this	Total	Including child / parent	Total	
	5.72	40.88	42.75	1	1	CaloTowersCreationAlgo::finish(edm::SortedCollection<CaloTower, edm:
[35]	5.72	1.59	39.29	1	1	**CaloTowersCreationAlgo::convert(CaloTowerDetId const&, CaloTowersCre**
	1.15	8.21	8.49	1	2	CaloTowerConstituentsMap::constituentsOf(CaloTowerDetId const&) cons
	0.98	7.00	7.01	1	2	EcalSeverityLevelAlgo::severityLevel(DetId const&, edm::SortedCollec
	0.61	4.33	4.39	1	2	void std::vector<CaloTower, std::allocator<CaloTower> >::_M_emplace_
	0.54	3.83	3.83	1	1	ROOT::Math::LorentzVector<ROOT::Math::PtEtaPhiM4D<double> >& ROOT::M
	0.36	2.55	2.55	1	1	CaloTowersCreationAlgo::emShwrPos(std::vector<std::pair<DetId, doubl
	0.31	2.21	2.21	1	1	reco::LeafCandidate::cacheCartesian() const [clone .part.137]
	0.30	2.14	37.70	1	990	free
	0.26	1.87	4.46	1	5	CaloTower::CaloTower(CaloTower const&)
	0.20	1.46	2.27	1	19	asinhf
	0.15	1.08	43.53	1	1,066	operator new(unsigned long)
	0.15	1.06	1.06	1	1	CaloTowersCreationAlgo::hadShwrPos(CaloTowerDetId, float)
	0.12	0.88	0.88	1	1	CaloTower::addConstituents(std::vector<DetId, std::allocator<DetId>
	0.10	0.74	6.22	1	31	cosh
	0.07	0.50	0.58	1	2	__gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<i
	0.06	0.43	0.43	1	2	Basic3DVector<float>::phi() const
	0.05	0.34	1.46	1	34	edm::HandleBase::productStorage() const
	0.03	0.21	0.21	1	1	Basic3DVector<float>::eta() const
	0.02	0.11	8.09	1	53	CaloSubdetectorGeometry::getGeometry(DetId const&) const
	0.01	0.10	2.06	1	4	void std::vector<std::pair<DetId, double>, std::allocator<std::pair<
	0.01	0.07	0.07	1	1	CaloTower::setCaloTowerStatus(unsigned int, unsigned int, unsigned i
	0.01	0.06	0.06	1	1	ROOT::Math::LorentzVector<ROOT::Math::PtEtaPhiM4D<double> >::Lorentz
	0.01	0.05	0.05	1	1	CaloTower::CaloTower(CaloTowerDetId const&, double, double, double,
	0.00	0.02	10.76	1	1,202	_init
	0.00	0.02	1.52	1	55	HcalDetId::HcalDetId(DetId const&)
	0.00	0.02	0.62	1	85	operator delete(void*)
	0.00	0.02	0.02	1	1	CaloTowersCreationAlgo::compactTime(float)
	0.00	0.01	0.09	1	8	reco::LeafCandidate::~LeafCandidate()

Back to summary

Typical effects of non trivial improvements



EcalRawToRecHitProducer/hltEcalRecHitAll

version	real time	cycles	ipc	br/ins	missed-br/cy	cache-ref/cy	mem-ref/cy	missed-L1I/cy	missed-L1D/cy	offcore/cy	div/cy	ind-call/cy	front-stall/cy	back-stall/cy	exec-stall/cy
original	307955943	1057418141	1.445	20.532	0.422	0.580	0.353	0.098	1.181	32.405	2.161	3.014	48.946	38.678	48.269
cleanup	246909037	845627941	1.420	17.518	0.511	0.702	0.447	0.091	1.211	43.769	2.738	1.290	54.644	43.648	51.363
prefetch	223314827	763354496	1.657	17.263	0.563	0.758	0.468	0.106	1.351	32.360	3.089	1.430	45.241	36.632	44.849
FastjetJetProducer	68687142964	2332148500	1.555	19.984	0.803	0.379	0.081	0.063	1.932	21.523	7.395	3.096	39.694	24.495	38.059
CkfTrackCandidateMaker	150365879112	522874051525	1.568	12.542	0.176	0.903	0.009	0.921	1.592	43.975	15.130	1.805	44.717	20.572	36.392
PFBlockProducer	2685989707	9307632673	2.973	23.901	0.186	0.380	0.018	0.007	4.533	12.275	0.327	3.961	15.724	6.236	16.858

Caller: loop over digi
Calls digi2hit functions (virtual)

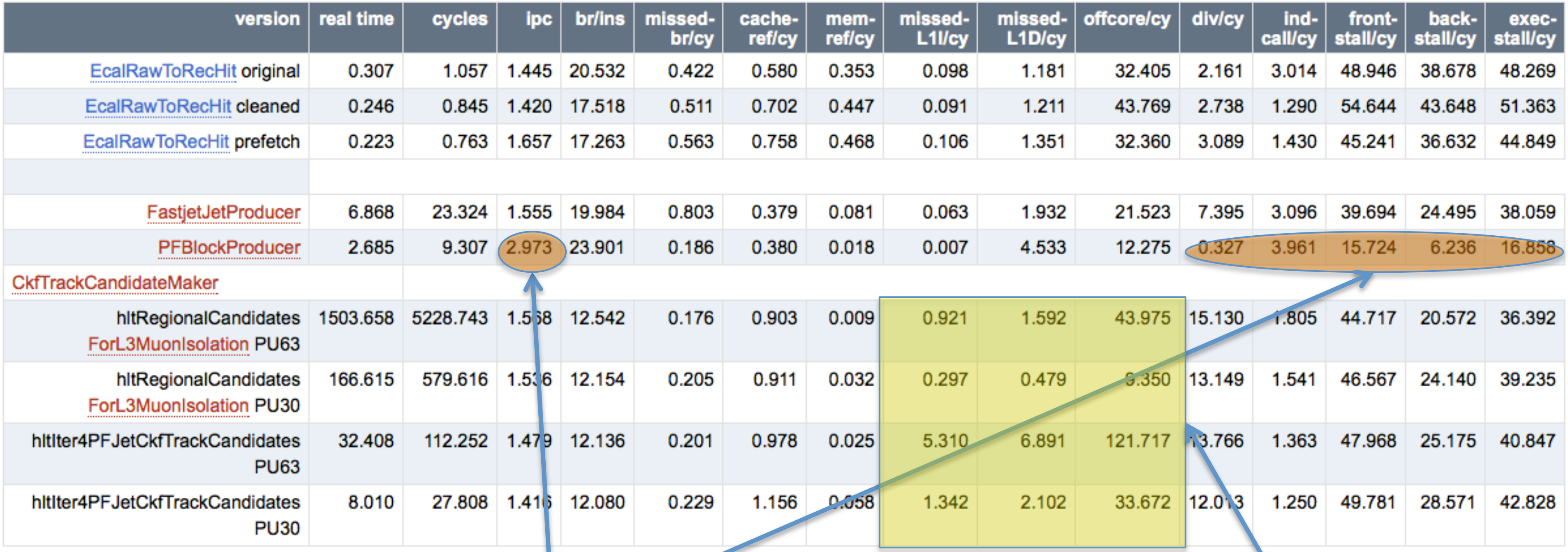
EcalUnpackerWorker.h:

```
namespace  bigHack {
  extern DetId nextOne;
}

 bigHack::nextOne =
((itdg+1)!=endDigi) ? DetId((itdg+1)->id()) : DetId();
```

Callee: applies calib to single digi

```
template<typename T>
inline void prefetchBarrel(const T & cond, int ind) {
   __builtin_prefetch(&cond.barrel(ind));
}
template<typename T>
inline void prefetchEndcap(const T & cond, int ind) {
   __builtin_prefetch(&cond.endcap(ind));
}

EcalUncalibRecHitWorkerWeightsOld::run( const edm::Event & evt,
            const EcalDigiCollection::const_iterator & itdg,
            EcalUncalibratedRecHitCollection & result ){
      DetId detid(itdg->id());
      auto next = bigHack::nextInd();
   prefetchEndcap(*peds,next);
   prefetchEndcap(*gains,next);
   prefetchEndcap(*grps,next);
```

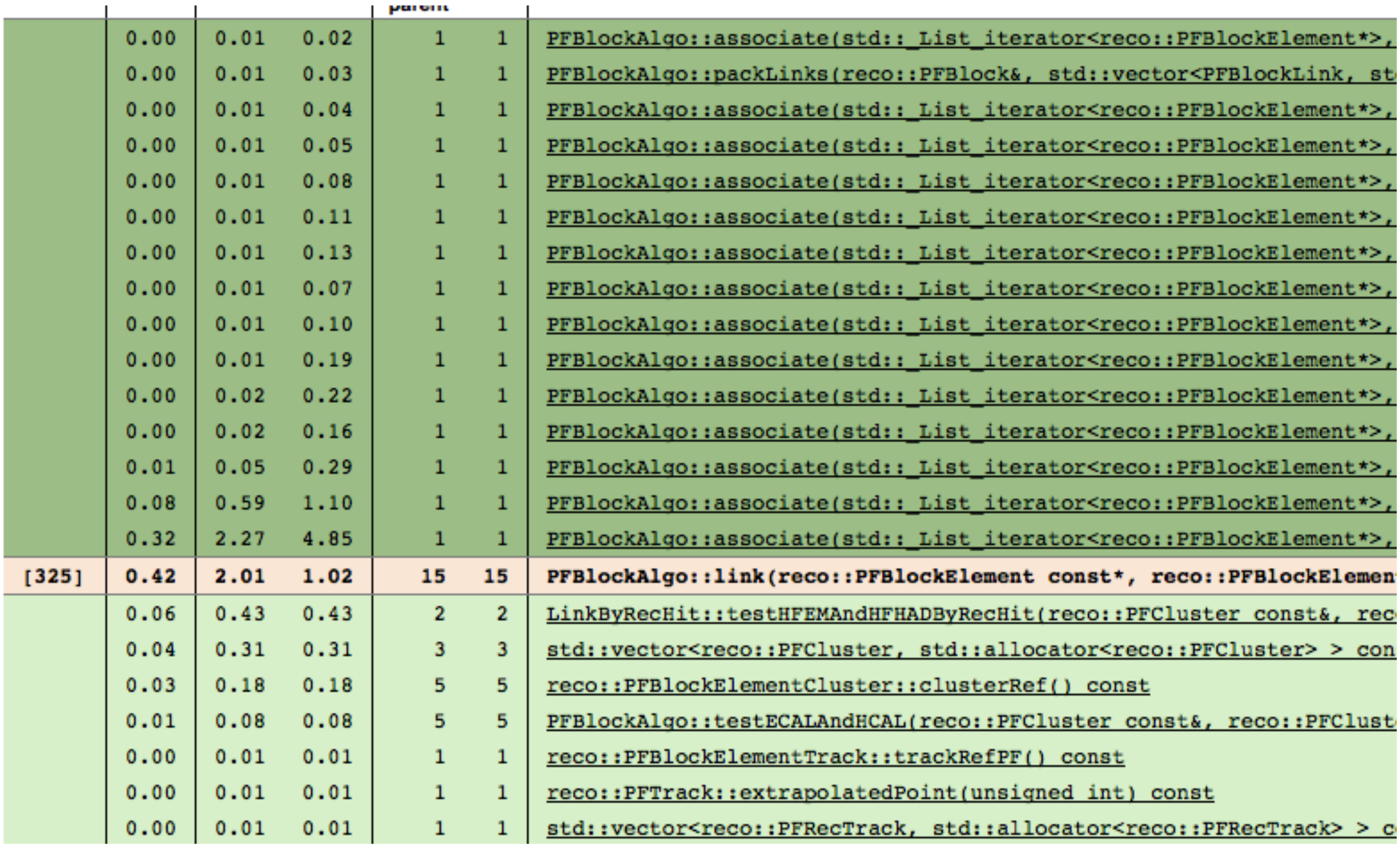
More examples



version	real time	cycles	ipc	br/ins	missed-br/cy	cache-ref/cy	mem-ref/cy	missed-L1I/cy	missed-L1D/cy	offcore/cy	div/cy	ind-call/cy	front-stall/cy	back-stall/cy	exec-stall/cy
EcalRawToRecHit original	0.307	1.057	1.445	20.532	0.422	0.580	0.353	0.098	1.181	32.405	2.161	3.014	48.946	38.678	48.269
EcalRawToRecHit cleaned	0.246	0.845	1.420	17.518	0.511	0.702	0.447	0.091	1.211	43.769	2.738	1.290	54.644	43.648	51.363
EcalRawToRecHit prefetch	0.223	0.763	1.657	17.263	0.563	0.758	0.468	0.106	1.351	32.360	3.089	1.430	45.241	36.632	44.849
FastjetJetProducer	6.868	23.324	1.555	19.984	0.803	0.379	0.081	0.063	1.932	21.523	7.395	3.096	39.694	24.495	38.059
PFBlockProducer	2.685	9.307	2.973	23.901	0.186	0.380	0.018	0.007	4.533	12.275	0.327	3.961	15.724	6.236	16.858
CkfTrackCandidateMaker															
hltRegionalCandidates ForL3MuonIsolation PU63	1503.658	5228.743	1.568	12.542	0.176	0.903	0.009	0.921	1.592	43.975	15.130	1.805	44.717	20.572	36.392
hltRegionalCandidates ForL3MuonIsolation PU30	166.615	579.616	1.536	12.154	0.205	0.911	0.032	0.297	0.479	9.350	13.149	1.541	46.567	24.140	39.235
hltIter4PFJetCkfTrackCandidates PU63	32.408	112.252	1.479	12.136	0.201	0.978	0.025	5.310	6.891	121.717	13.766	1.363	47.968	25.175	40.847
hltIter4PFJetCkfTrackCandidates PU30	8.010	27.808	1.416	12.080	0.229	1.156	0.058	1.342	2.102	33.672	12.013	1.250	49.781	28.571	42.828

Recursive Algorithm to build a NN graph.
Huge case switch + condition on “distance”.

Some code,
Different “conditions”



				parent		
	0.00	0.01	0.02	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.01	0.03	1	1	PFBlockAlgo::packLinks(reco::PFBlock&, std::vector<PFBlockLink, st
	0.00	0.01	0.04	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.01	0.05	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.01	0.08	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.01	0.11	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.01	0.13	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.01	0.07	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.01	0.10	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.01	0.19	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.02	0.22	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.00	0.02	0.16	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.01	0.05	0.29	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.08	0.59	1.10	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
	0.32	2.27	4.85	1	1	PFBlockAlgo::associate(std::_List_iterator<reco::PFBlockElement*>,
[325]	**0.42**	**2.01**	**1.02**	**15**	**15**	**PFBlockAlgo::link(reco::PFBlockElement const*, reco::PFBlockElemen**
	0.06	0.43	0.43	2	2	LinkByRecHit::testHFEMAndHFHADByRecHit(reco::PFCluster const&, rec
	0.04	0.31	0.31	3	3	std::vector<reco::PFCluster, std::allocator<reco::PFCluster> > con
	0.03	0.18	0.18	5	5	reco::PFBlockElementCluster::clusterRef() const
	0.01	0.08	0.08	5	5	PFBlockAlgo::testECALAndHCAL(reco::PFCluster const&, reco::PFClust
	0.00	0.01	0.01	1	1	reco::PFBlockElementTrack::trackRefPF() const
	0.00	0.01	0.01	1	1	reco::PFTrack::extrapolatedPoint(unsigned int) const
	0.00	0.01	0.01	1	1	std::vector<reco::PFRecTrack, std::allocator<reco::PFRecTrack> > c

Non-trivial optimization opportunities

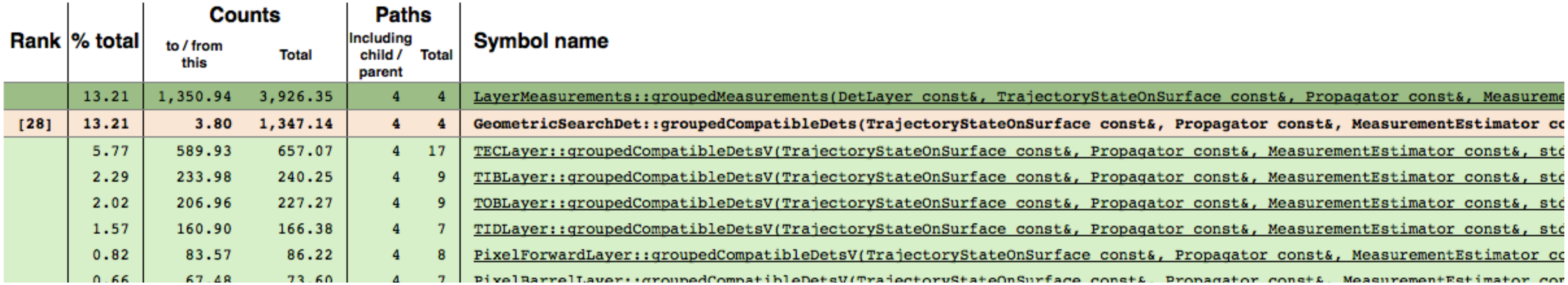
- "amortize" or remove cost of divisions/sqrt
- enable vectorization
- Reduce cost of indirect and virtual calls
- Software prefetch of calibration constants
- Reduce caching of unused quantities
- Match precision to target accuracy
- New challenges:
  - Thread safety, thread friendness, high-granularity parallelism

Need Tools to help identify issues and guide fixes

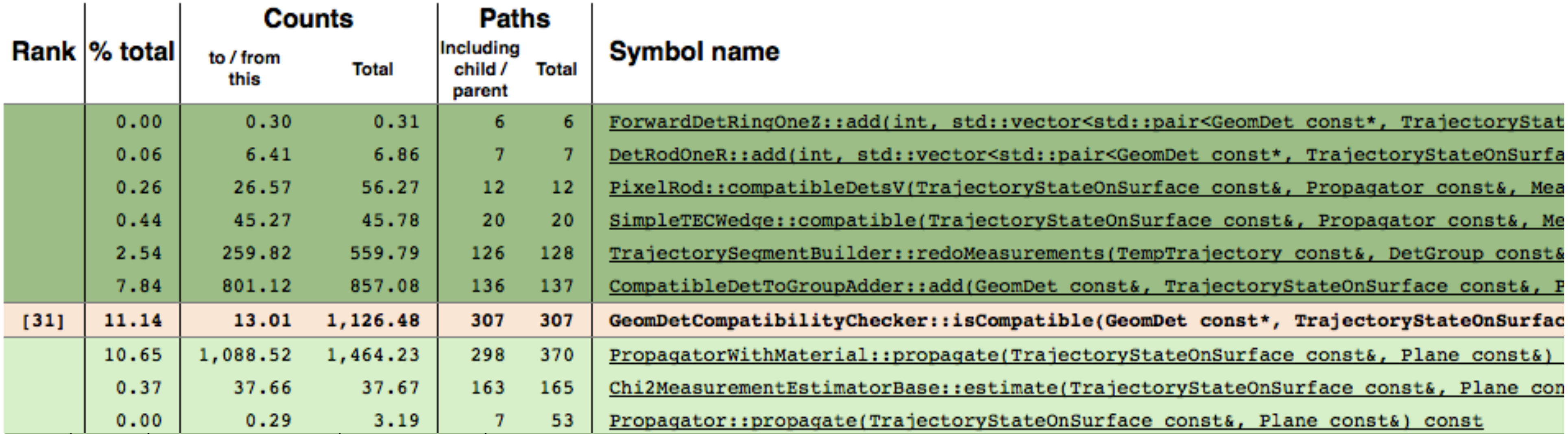
Difficult Profiling Challenges

- Same code may behave very differently when dealing with different data or parameters
  - Best optimization solution could be to factorize code and/or use different data structures:
    - need to identify various components first!
  - Need "user defined" mechanism to tag "counts" coming from these different usages in the same job
    - Phases in iterations
    - Different geometrical origin

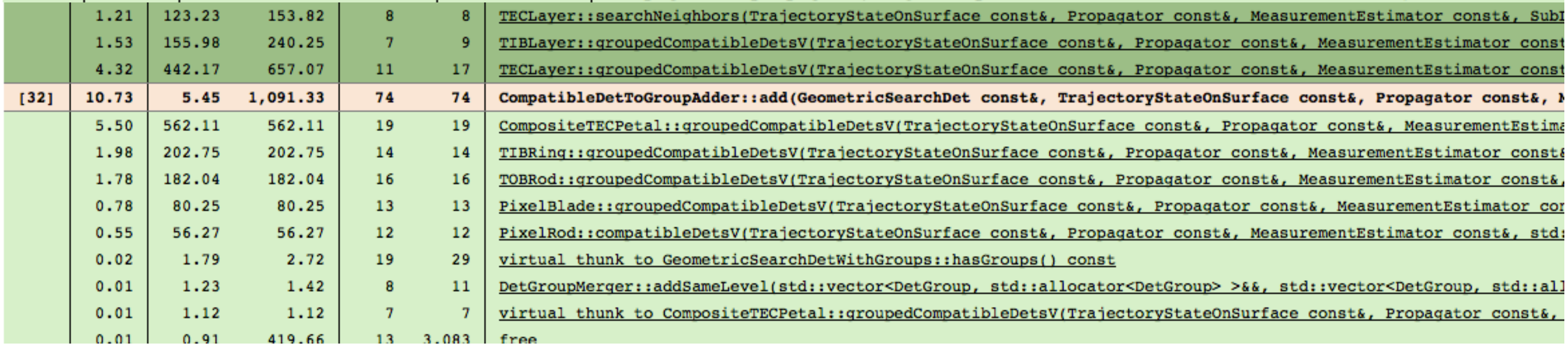
Complex Diamonds



Rank	% total	Counts		Paths		Symbol name
		to / from this	Total	Including child / parent	Total	
	13.21	1,350.94	3,926.35	4	4	LayerMeasurements::groupedMeasurements(DetLayer const&, TrajectoryStateOnSurface const&, Propagator const&, Measureme
[28]	**13.21**	**3.80**	**1,347.14**	**4**	**4**	**GeometricSearchDet::groupedCompatibleDets(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator co**
	5.77	589.93	657.07	4	17	TECLayer::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const&, std
	2.29	233.98	240.25	4	9	TIBLayer::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const&, std
	2.02	206.96	227.27	4	9	TOBLayer::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const&, std
	1.57	160.90	166.38	4	7	TIDLayer::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const&, std
	0.82	83.57	86.22	4	8	PixelForwardLayer::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator co
	0.66	67.48	73.60	4	7	PixelBarrelLayer::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator cor



Rank	% total	Counts		Paths		Symbol name
		to / from this	Total	Including child / parent	Total	
	0.00	0.30	0.31	6	6	ForwardDetRingOneZ::add(int, std::vector<std::pair<GeomDet const*, TrajectoryStat
	0.06	6.41	6.86	7	7	DetRodOneR::add(int, std::vector<std::pair<GeomDet const*, TrajectoryStateOnSurfa
	0.26	26.57	56.27	12	12	PixelRod::compatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, Mea
	0.44	45.27	45.78	20	20	SimpleTECWedge::compatible(TrajectoryStateOnSurface const&, Propagator const&, Me
	2.54	259.82	559.79	126	128	TrajectorySegmentBuilder::redoMeasurements(TempTrajectory const&, DetGroup const&
	7.84	801.12	857.08	136	137	CompatibleDetToGroupAdder::add(GeomDet const&, TrajectoryStateOnSurface const&, P
[31]	**11.14**	**13.01**	**1,126.48**	**307**	**307**	**GeomDetCompatibilityChecker::isCompatible(GeomDet const*, TrajectoryStateOnSurfac**
	10.65	1,088.52	1,464.23	298	370	PropagatorWithMaterial::propagate(TrajectoryStateOnSurface const&, Plane const&)
	0.37	37.66	37.67	163	165	Chi2MeasurementEstimatorBase::estimate(TrajectoryStateOnSurface const&, Plane con
	0.00	0.29	3.19	7	53	Propagator::propagate(TrajectoryStateOnSurface const&, Plane const&) const

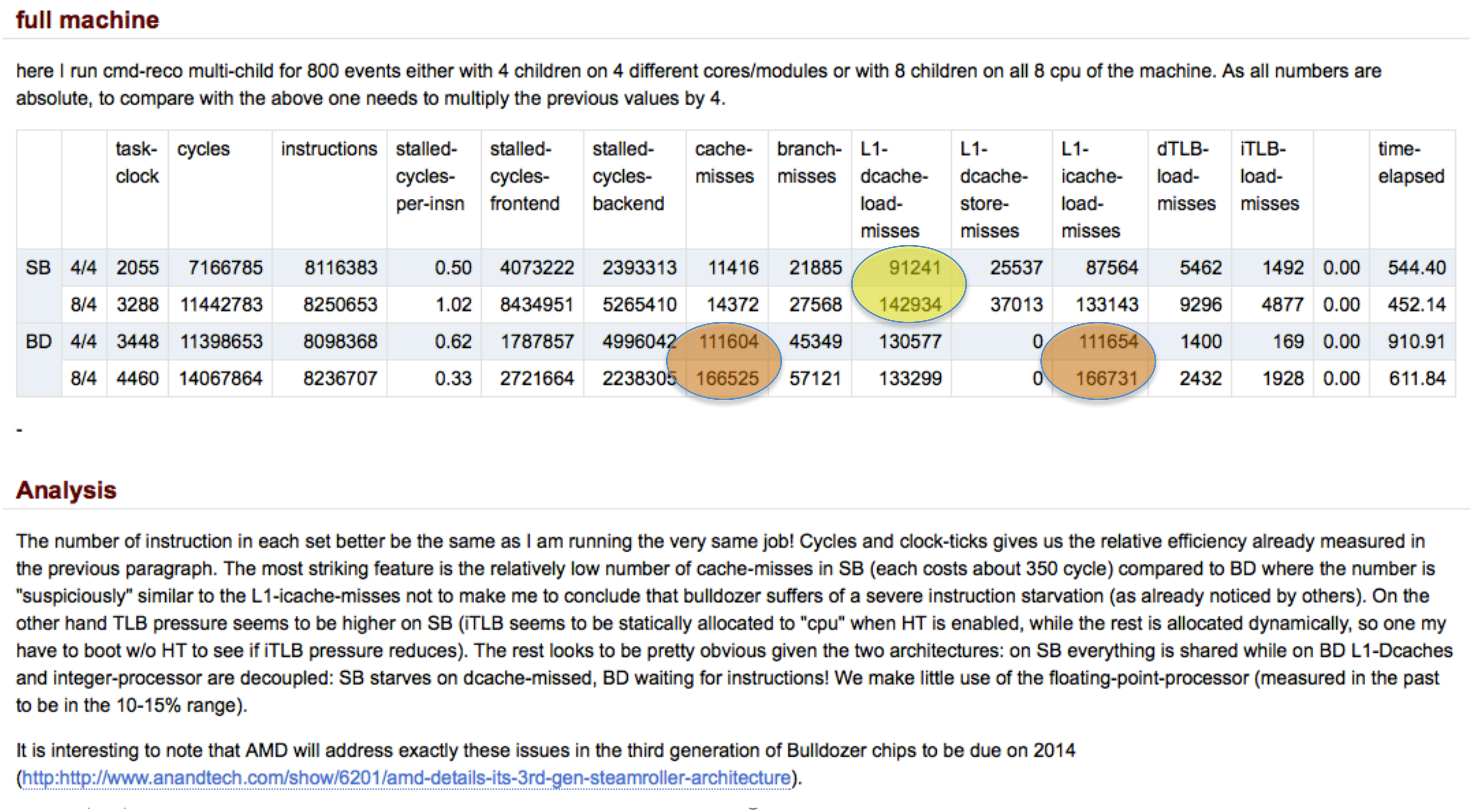


	1.21	123.23	153.82	8	8	TECLayer::searchNeighbors(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const&, SubI
	1.53	155.98	240.25	7	9	TIBLayer::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const
	4.32	442.17	657.07	11	17	TECLayer::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const
[32]	**10.73**	**5.45**	**1,091.33**	**74**	**74**	**CompatibleDetToGroupAdder::add(GeometricSearchDet const&, TrajectoryStateOnSurface const&, Propagator const&, M**
	5.50	562.11	562.11	19	19	CompositeTECPetal::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstima
	1.98	202.75	202.75	14	14	TIBRing::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const&
	1.78	182.04	182.04	16	16	TOBRod::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const&,
	0.78	80.25	80.25	13	13	PixelBlade::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator cor
	0.55	56.27	56.27	12	12	PixelRod::compatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&, MeasurementEstimator const&, std:
	0.02	1.79	2.72	19	29	virtual thunk to GeometricSearchDetWithGroups::hasGroups() const
	0.01	1.23	1.42	8	11	DetGroupMerger::addSameLevel(std::vector<DetGroup, std::allocator<DetGroup> >&&, std::vector<DetGroup, std::all
	0.01	1.12	1.12	7	7	virtual thunk to CompositeTECPetal::groupedCompatibleDetsV(TrajectoryStateOnSurface const&, Propagator const&,
	0.01	0.91	419.66	13	3,083	free

More use cases

- Understand differences among architectures
  - Support to Benchmark
  - Optimization by platform
- Understand concurrency scaling
  - Cache hierarchy
  - Threads (HT)
  - Numa

SandyBridge vs Buldozer (using perf)



full machine

here I run cmd-reco multi-child for 800 events either with 4 children on 4 different cores/modules or with 8 children on all 8 cpu of the machine. As all numbers are absolute, to compare with the above one needs to multiply the previous values by 4.

		task-clock	cycles	instructions	stalled-cycles-per-insn	stalled-cycles-frontend	stalled-cycles-backend	cache-misses	branch-misses	L1-dcache-load-misses	L1-dcache-store-misses	L1-icache-load-misses	dTLB-load-misses	iTLB-load-misses		time-elapsed
SB	4/4	2055	7166785	8116383	0.50	4073222	2393313	11416	21885	91241	25537	87564	5462	1492	0.00	544.40
	8/4	3288	11442783	8250653	1.02	8434951	5265410	14372	27568	142934	37013	133143	9296	4877	0.00	452.14
BD	4/4	3448	11398653	8098368	0.62	1787857	4996042	111604	45349	130577	0	111654	1400	169	0.00	910.91
	8/4	4460	14067864	8236707	0.33	2721664	2238305	166525	57121	133299	0	166731	2432	1928	0.00	611.84

-

Analysis

The number of instruction in each set better be the same as I am running the very same job! Cycles and clock-ticks gives us the relative efficiency already measured in the previous paragraph. The most striking feature is the relatively low number of cache-misses in SB (each costs about 350 cycle) compared to BD where the number is "suspiciously" similar to the L1-icache-misses not to make me to conclude that bulldozer suffers of a severe instruction starvation (as already noticed by others). On the other hand TLB pressure seems to be higher on SB (iTLB seems to be statically allocated to "cpu" when HT is enabled, while the rest is allocated dynamically, so one my have to boot w/o HT to see if iTLB pressure reduces). The rest looks to be pretty obvious given the two architectures: on SB everything is shared while on BD L1-Dcaches and integer-processor are decoupled: SB starves on dcache-missed, BD waiting for instructions! We make little use of the floating-point-processor (measured in the past to be in the 10-15% range).

It is interesting to note that AMD will address exactly these issues in the third generation of Bulldozer chips to be due on 2014 (http:http://www.anandtech.com/show/6201/amd-details-its-3rd-gen-steamroller-architecture).

Haswell (sse4.2)

data statically partitioned among threads

```
 Performance counter stats for
'./main_XEON -n 600000 -b 512 -c -i 200':

 8 cpu-migrations   #   0.000 K/sec
84885.733468 task-clock      #   3.993 CPUs utilized
293801901385 cycles          #   3.461 GHz
264499508080 instructions    #   0.90  insns per cycle
223258439563 r180010e        # 2630.106 M/sec
156260450712 r180ffa1        # 1840.833 M/sec
  1574547747 cache-misses    #  55.696 % of all cache refs
  2827037599 cache-references #  33.304 M/sec
  9658461968 branches        #  113.782 M/sec
    46088085 branch-misses   #   0.48% of all branches
  6057883186 r0151           #   71.365 M/sec
    20073044 r0280           #   0.236 M/sec
         279 r0408           #   0.003 K/sec
          27 r0485           #   0.000 K/sec

21.257407715 seconds time elapsed
```

data shared, tasks scheduled to maximize reuse

```
Performance counter stats for
'./main_XEON -n 600000 -b 512 -s -i 200':

           5 cpu-migrations  #   0.000 K/sec
36586.081199 task-clock      #   3.988 CPUs utilized
127243639023 cycles          #   3.478 GHz
242793532688 instructions    #   1.91  insns per cycle
  63033060764 r180010e       # 1722.870 M/sec
  18619353186 r180ffa1       #  508.919 M/sec
  168155515 cache-misses     #   41.999 % of all cache refs
  400383640 cache-references #  10.944 M/sec
 9380022273 branches         #  256.382 M/sec
   55000265 branch-misses    #   0.59% of all branches
 5441062294 r0151            #  148.719 M/sec
    7432557 r0280            #   0.203 M/sec
        876 r0408            #   0.024 K/sec
         90 r0485            #   0.002 K/sec

9.173927783 seconds time elapsed
```

Haswell (avx2)

data statically partitioned among threads

```
 Performance counter stats for
'./main_XEON -n 600000 -b 512 -c -i 200':

  6 cpu-migrations           #    0.000 K/sec
 83270.337342 task-clock              #    3.993 CPUs utilized
287867948317 cycles                  #    3.457 GHz
124797441793 instructions         #    0.43  insns per cycle
250537013069 r180010e            # 3008.719 M/sec
207507514834 r180ffa1          # 2491.974 M/sec
1822959632 cache-misses        #    58.744 % of all cache refs
3103202150 cache-references        #   37.267 M/sec
     5822378687 branches           #   69.921 M/sec
     26932431 branch-misses        #    0.46% of all branches
     6032047991 r0151              #   72.439 M/sec
       19434992 r0280              #    0.233 M/sec
           1071 r0408            #    0.013 K/sec
             78 r0485            #    0.001 K/sec

  20.854578941 seconds time elapsed
```

data shared, tasks scheduled to maximize reuse

```
Performance counter stats for
'./main_XEON -n 600000 -b 512 -s -i 200':

 7 cpu-migrations           #    0.000 K/sec
27003.862077 task-clock              #    3.986 CPUs utilized
 93692283234 cycles                  #    3.470 GHz
115577324193 instructions           #    1.23  insns per cycle
 59912282888 r180010e              # 2218.656 M/sec
 31434453581 r180ffa1             # 1164.073 M/sec
 284980434 cache-misses         #    54.079 % of all cache refs
 526967712 cache-references          #   19.515 M/sec
  5650405685 branches              #  209.244 M/sec
    37149664 branch-misses           #    0.66% of all branches
  5463010488 r0151                 #  202.305 M/sec
     5411827 r0280               #    0.200 M/sec
        1540 r0408             #    0.057 K/sec
          33 r0485             #    0.001 K/sec

 6.775450738 seconds time elapsed
```

Silvermont

data statically partitioned among threads

```
Performance counter stats for
'./main_XEON -n 600000 -b 512 -c -i 200':

        17 cpu-migrations            #    0.000 K/sec
  447603.908846 task-clock           #    7.964 CPUs utilized
753,155,749,740 cycles               #    1.683 GHz
265,846,847,979 instructions         #    0.35  insns per cycle
753,693,090,985 r180010e             # 1683.839 M/sec
753,910,391,255 r180ffa1             # 1684.325 M/sec
  4,749,770,725 cache-misses         #   35.796 % of all cache refs
 13,268,896,593 cache-refs           #   29.644 M/sec
 10,120,308,666 branches             #   22.610 M/sec
    507,413,900 branch-misses        #    5.01% of all branches
              0 r0151                #    0.000 K/sec
     80,070,430 r0280                #    0.179 M/sec
              0 r0408                #    0.000 K/sec
              0 r0485                #    0.000 K/sec

 56.204619501 seconds time elapsed
```

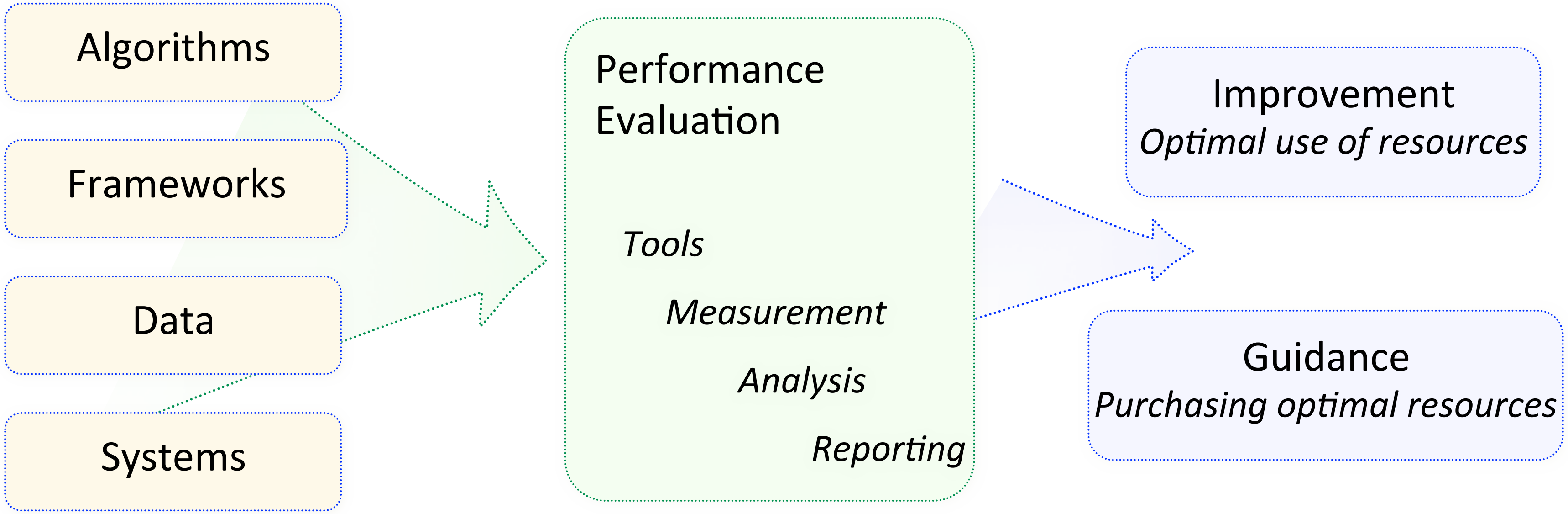
data shared, tasks scheduled to maximize reuse

```
Performance counter stats for
'./main_XEON -n 600000 -b 512 -s -i 200':

        11 cpu-migrations               #    0.000 K/sec
  397260.599177 task-clock              #    7.963 CPUs utilized
668,800,742,409 cycles                  #    1.684 GHz
243,590,222,327 instructions            #    0.36  insns per cycle
669,372,104,392 r180010e                # 1684.970 M/sec
669,673,854,601 r180ffa1                # 1685.729 M/sec
    764,128,803 cache-misses            #    9.559 % of all cache refs
  7,993,691,948 cache-references        #   20.122 M/sec
  9,716,850,290 branches                #   24.460 M/sec
    510,405,714 branch-misses           #    5.25% of all branches
              0 r0151                   #    0.000 K/sec
     89,028,935 r0280                   #    0.224 M/sec
              0 r0408                   #    0.000 K/sec
              0 r0485                   #    0.000 K/sec

 49.887540985 seconds time elapsed
```

Summary

- Large Complex code
- Multi-facet complex behavior
- Many developers (doing mostly science)
- Few experts (doing mostly management)

Need tools to foster performance data, analyze them and provide guidance toward the most efficient optimization actions



Algorithms
Frameworks
Data
Systems
Performance Evaluation
Tools
Measurement
Analysis
Reporting
Improvement
Optimal use of resources
Guidance
Purchasing optimal resources