Table of Contents

Nu, Mu and Pile-Up. The LHCb definitions of what we see and what we don't see.

LHC luminosity calculator

See LHC luminosity calculator.

(nu) : The average number of pp interactions per bunch crossing

This includes elastic and diffractive (at 7 TeV that would be about 91 mb). This is the number we use in the simulation as input to pythia. MC09 and 2010 MC have been generated with =1. The breakdown of subprocesses according to pythia is :

Interaction	cross-section
total	91.05 mb
elastic	19.38 mb
single diffractive	2 x 6.84 mb
double diffractive	9.19 mb
inelastic = total - everything else	48.80 mb
inelastic = total - elastic only	71.67 mb

It is not completely clear which fraction of this is visible. Hans ran L0 with TCK 1810 on 2010 MC single-interaction events and gets an L0-accept of 69.9%. This would mean the *visible* fraction is 63.6 mb. This is somewhere in between "total - everything else" and "total - elastic only". And of course no one knows what the actual cross sections are. Totem should tell us.

In the simulation we store only events with at least one pp interaction. The mean number of interactions per event is thus about 1.3. Yet about 20% of the events have no hits in the detector. They are purely elastic events.

μ (mu): The average number of visible pp interactions per bunch crossing

This is the only thing we can measure at the pit. Hence all numbers coming from the pit should be understood as μ. Until we know more, we will assume this is the fraction 69.9% described above, i.e. μ = 0.699 .

The formula for μ from the instantaneous luminosity and number of bunches is: $\mu = L_{tot}/N_{bunches}/0.177\ 10^{30}$. For L_{tot}=4.0 10^{32} and $N_{bunches}$=1262, μ=1.79 (at 7 TeV, a bit less at 8 TeV).

Pile-up: the average number of pp interactions in visible events

This is the relevant quantity for the Hlt processing time. Assuming a L0 trigger selecting all visible interactions, we send to the event filter farm all events that contain at least one visible pp interaction. The Poisson distribution for a mean μ is $P(n) = \mu^n \exp(-\mu)/n!$. The pile-up is the zero-suppressed mean of this distribution = $\mu/(1-P(0)) = \mu/(1-\exp(-\mu))$. At μ=1 this becomes 1.58. For μ=0.7 it is 1.38.

We cannot actually measure this quantity directly. What we measure is P(0)=exp(-μ) from which we infer μ and thus the pile-up.

Note that this average pile-up is not what we see in the Hlt if we have a biasing L0. It gets event worse after Hlt1 and Hlt2. It is thus not a good estimate of the average event size coming out of the pit.

Conclusion

When talking about real data, always use µ. is only useful to compare with MC.

Appendix 1 : Find out the mu of a run

You can find it out yourself for every fill using the recipe given in MuFromFill.

For the September-October 2010 data PK made a dictionary of run at beginning of run vs run, based on the luminosity reported in the e-log:

```
muAtBeginOfRun=  { 81685: 1.22, 81684: 1.31, 81683: 1.37, 81680: 1.46, 81679: 1.61, 81678: 1.67,
81620: 1.28, 81619: 1.34, 81618: 1.36, 81616: 1.40, 81615: 1.46, 81614: 1.48, 81613: 1.48, 81611:
81602: 1.87, 81601: 1.90, 81600: 1.96, 81599: 2.08, 81598: 2.08, 81597: 2.15, 81596: 2.27, 81595:
81468: 2.05, 81467: 2.09, 81466: 2.09, 81465: 2.22, 81464: 2.27, 81463: 2.31, 81462: 2.39, 81461:
81374: 1.53, 81373: 1.54, 81372: 1.57, 81371: 1.62, 81370: 1.66, 81369: 1.66, 81368: 1.72, 81367:
81358: 2.17, 81357: 2.31, 81356: 2.33, 81355: 2.36, 81353: 2.52, 81352: 2.60, 81351: 2.61, 81350:
81322: 1.68, 81321: 1.75, 81320: 1.84, 81317: 1.86, 81316: 1.87, 81315: 1.97, 81314: 2.08, 81313:
80848: 0.95, 80847: 0.99, 80846: 1.04, 80845: 1.08, 80844: 1.12, 80843: 1.16, 80842: 1.21, 80841:
80696: 1.32, 80695: 1.47, 80693: 1.54, 80692: 1.56, 80691: 1.66, 80690: 1.72, 80689: 1.74, 80688:
80588: 1.47, 80587: 1.54, 80586: 1.66, 80585: 1.75, 80584: 1.85, 80578: 1.93, 80577: 1.96, 80405:
80395: 1.92, 80394: 1.95, 80393: 1.95, 80392: 1.96, 80360: 1.43, 80357: 1.49, 80356: 1.55, 80355:
80234: 1.01, 80233: 1.01, 80232: 1.05, 80231: 1.09, 80229: 1.11, 80228: 1.12, 80227: 1.16, 80226:
80214: 1.68, 80213: 1.68, 80212: 1.68, 80211: 1.71, 80210: 1.72, 80209: 1.73, 80173: 1.40, 80172:
80163: 1.81, 80162: 1.83, 80161: 1.88, 80160: 1.91, 80085: 1.13, 80084: 1.15, 80083: 1.16, 80082:
80069: 1.43, # this was reverse-engineered from the integrated luminosity. The Lumi reported in t
80033: 0.10, 80032: 0.10, 80030: 0.11, 80029: 0.11, 79951: 1.66,
79950: 1.81, 79949: 1.81, 79886: 1.39, 79885: 1.45, 79884: 1.51, 79883: 1.53, 79882: 1.62, 79881:
79839: 1.29, 79838: 1.37, 79837: 1.43, 79836: 1.47, 79835: 1.54, 79834: 1.55, 79833: 1.56, 79832:
79802: 0.72, 79801: 0.72, 79795: 0.77, 79794: 0.82, 79793: 0.82,
79792: 0.88, 79791: 0.94, 79790: 1.06, 79789: 1.12, 79788: 1.19, 79787: 1.29, 79786: 1.41, 79785:
79753: 2.05, 79660: 1.19, 79659: 1.25, 79658: 1.32, 79657: 1.34, 79656: 1.41, 79655: 1.49, 79654:
79586: 0.96, 79585: 1.00, 79584: 1.03, 79583: 1.07, 79582: 1.10, 79581: 1.13, 79580: 1.15, 79579:
```

This can be passed to `DecayTreeTuple` as

```
from Configurables import TupleToolEventInfo
tuple.addTool(TupleToolEventInfo)
tuple.TupleToolEventInfo.Mu = muAtBeginOfRun
```

or pasted directly in any data structure...

Appendix 2 : a huge table

(nu)	mu	PU	0 pp			1 pp			2 pp			3 pp			4 pp	
			P(0)	P(0|Vis)	Lumi	P(1)	P(0|Vis)	Lumi	P(2)	P(2|Vis)	Lumi	P(3)	P(3|Vis)	Lumi	P(4)	P(4|Vis)
0.07	**0.05**	1.03	95%	0%	0%	5%	98%	95%	0%	2%	5%	0%	0%	0%	0%	0%
0.14	**0.10**	1.05	90%	0%	0%	9%	95%	90%	0%	5%	9%	0%	0%	0%	0%	0%
0.21	**0.15**	1.08	86%	0%	0%	13%	93%	86%	1%	7%	13%	0%	0%	1%	0%	0%
0.29	**0.20**	1.10	82%	0%	0%	16%	90%	82%	2%	9%	16%	0%	1%	2%	0%	0%
0.36	**0.25**	1.13	78%	0%	0%	19%	88%	78%	2%	11%	19%	0%	1%	2%	0%	0%
0.43	**0.30**	1.16	74%	0%	0%	22%	86%	74%	3%	13%	22%	0%	1%	3%	0%	0%

0.50	**0.35**	1.19	70%	0%	0%	25%	84%	70%	4%	15%	25%	1%	2%	4%	0%	0%
0.57	**0.40**	1.21	67%	0%	0%	27%	81%	67%	5%	16%	27%	1%	2%	5%	0%	0%
0.64	**0.45**	1.24	64%	0%	0%	29%	79%	64%	6%	18%	29%	1%	3%	6%	0%	0%
0.72	**0.50**	1.27	61%	0%	0%	30%	77%	61%	8%	19%	30%	1%	3%	8%	0%	0%
0.79	**0.55**	1.30	58%	0%	0%	32%	75%	58%	9%	21%	32%	2%	4%	9%	0%	1%
0.86	**0.60**	1.33	55%	0%	0%	33%	73%	55%	10%	22%	33%	2%	4%	10%	0%	1%
0.93	**0.65**	1.36	52%	0%	0%	34%	71%	52%	11%	23%	34%	2%	5%	11%	0%	1%
1.00	**0.70**	1.39	50%	0%	0%	35%	69%	50%	12%	24%	35%	3%	6%	12%	0%	1%
1.07	**0.75**	1.42	47%	0%	0%	35%	67%	47%	13%	25%	35%	3%	6%	13%	1%	1%
1.14	**0.80**	1.45	45%	0%	0%	36%	65%	45%	14%	26%	36%	4%	7%	14%	1%	1%
1.22	**0.85**	1.48	43%	0%	0%	36%	63%	43%	15%	27%	36%	4%	8%	15%	1%	2%
1.29	**0.90**	1.52	41%	0%	0%	37%	62%	41%	16%	28%	37%	5%	8%	16%	1%	2%
1.36	**0.95**	1.55	39%	0%	0%	37%	60%	39%	17%	28%	37%	6%	9%	17%	1%	2%
1.43	**1.00**	1.58	37%	0%	0%	37%	58%	37%	18%	29%	37%	6%	10%	18%	2%	2%
1.57	**1.10**	1.65	33%	0%	0%	37%	55%	33%	20%	30%	37%	7%	11%	20%	2%	3%
1.72	**1.20**	1.72	30%	0%	0%	36%	52%	30%	22%	31%	36%	9%	12%	22%	3%	4%
1.86	**1.30**	1.79	27%	0%	0%	35%	49%	27%	23%	32%	35%	10%	14%	23%	3%	4%
2.00	**1.40**	1.86	25%	0%	0%	35%	46%	25%	24%	32%	35%	11%	15%	24%	4%	5%
2.15	**1.50**	1.93	22%	0%	0%	33%	43%	22%	25%	32%	33%	13%	16%	25%	5%	6%
2.29	**1.60**	2.00	20%	0%	0%	32%	40%	20%	26%	32%	32%	14%	17%	26%	6%	7%
2.43	**1.70**	2.08	18%	0%	0%	31%	38%	18%	26%	32%	31%	15%	18%	26%	6%	8%
2.58	**1.80**	2.16	17%	0%	0%	30%	36%	17%	27%	32%	30%	16%	19%	27%	7%	9%
2.72	**1.90**	2.23	15%	0%	0%	28%	33%	15%	27%	32%	28%	17%	20%	27%	8%	10%
2.86	**2.00**	2.31	14%	0%	0%	27%	31%	14%	27%	31%	27%	18%	21%	27%	9%	10%
3.22	**2.25**	2.52	11%	0%	0%	24%	27%	11%	27%	30%	24%	20%	22%	27%	11%	13%
3.58	**2.50**	2.72	8%	0%	0%	21%	22%	8%	26%	28%	21%	21%	23%	26%	13%	15%
3.93	**2.75**	2.94	6%	0%	0%	18%	19%	6%	24%	26%	18%	22%	24%	24%	15%	16%
4.29	**3.00**	3.16	5%	0%	0%	15%	16%	5%	22%	24%	15%	22%	24%	22%	17%	18%
4.65	**3.25**	3.38	4%	0%	0%	13%	13%	4%	20%	21%	13%	22%	23%	20%	18%	19%
5.01	**3.50**	3.61	3%	0%	0%	11%	11%	3%	18%	19%	11%	22%	22%	18%	19%	19%
5.36	**3.75**	3.84	2%	0%	0%	9%	9%	2%	17%	17%	9%	21%	21%	17%	19%	20%
5.72	**4.00**	4.07	2%	0%	0%	7%	7%	2%	15%	15%	7%	20%	20%	15%	20%	20%
6.08	**4.25**	4.31	1%	0%	0%	6%	6%	1%	13%	13%	6%	18%	19%	13%	19%	20%
6.44	**4.50**	4.55	1%	0%	0%	5%	5%	1%	11%	11%	5%	17%	17%	11%	19%	19%
6.80	**4.75**	4.79	1%	0%	0%	4%	4%	1%	10%	10%	4%	15%	16%	10%	18%	19%
7.15	**5.00**	5.03	1%	0%	0%	3%	3%	1%	8%	8%	3%	14%	14%	8%	18%	18%

For each mu and number of visible pp collisions, the table gives the probability as it comes out of the Poisson law (P(n)), the faction among non-empty events (P(n|Vis)), and the probability multiplied by the number of interactions (lumi), n*P(n).

The **visible fraction** is 1-P(0). So for μ=1, 63% of the events are visible. The frequency of the LHC being 11 kHz that means that the **visible rate** is 0.63*11 kHz = 7 kHz per pair of colliding bunches. This rate only increases slowly with μ.

To find out the value of mu for a given data, look here.

-- PatrickSKoppenburg - 12-Jul-2010

--- This topic: LHCb > NuMuPileUp
Topic revision: r20 - 2017-06-06 - BenjaminCouturier