

# CBC, Latches, and Response Time

Author: Craig Shallahamer (craig@orapub.com), Version 1d, 14-July-2011

## Background and Purpose

*The purpose of this notepad is to see if increasing CBC latches can decrease CPU and non-idle wait time per buffer get.*

## Experimental Data

Below is all the experimental data. The experiment was run on a Dell single four-core CPU, Oracle 11.2G. According to “cat /proc/version”: Linux version 2.6.18-164.el5PAE (mockbuild@ca-build10.us.oracle.com) (gcc version 4.1.2 20080704 (Red Hat 4.1.2-46)) #1 SMP Thu Sep 3 02:28:20 EDT 2009. There was a tremendous CBC latch contention load, the OS was CPU bottlenecked at 100% utilization and the CPU run queue always at least 3X greater than the number of CPU cores (4). For each sample set, 90 samples were collected each with an interval of 180 seconds. There was also a calm time of 4 seconds between each sample. When the number of latches was changed, the instance was (obviously) cycled, the number of latched checked, the identical CBC load restarted, and the system was allowed to stabilize for five minutes before the sample collection began.

The order of sample data is sample number, elapsed time (seconds), buffer gets (session logical reads), instance non-idle wait time (sec), instance CPU consumption (sec), and DB time (sec).

ln[1]:=

```
ss1024 = {1, 180.806271, 80343366, 656.72, 718.807724, 2253.757892, 2, 180.610454, 80335763, 651.6,
715.975149, 2240.941899, 3, 180.645746, 80471221, 656.3, 719.223654, 2255.543044, 4, 180.675562,
80892542, 670.81, 719.535614, 2251.484661, 5, 181.18038, 80902024, 669.25, 722.501153,
2271.511149, 6, 185.147278, 80841438, 679.84, 735.530179, 2319.304952, 7, 182.14873, 80886294,
679.66, 724.983779, 2287.894825, 8, 181.978054, 80635492, 651.05, 726.28859, 2275.954513,
9, 182.368228, 80495691, 660.38, 727.426403, 2275.274329, 10, 180.527711, 81547482, 663.04,
716.103136, 2251.788529, 11, 180.249626, 80448750, 670.08, 718.924698, 2255.679078, 12,
180.248411, 80385375, 674.18, 715.793176, 2246.736735, 13, 182.094783, 81376146, 643.88,
722.756119, 2269.004708, 14, 181.937686, 80768004, 668.47, 726.54054, 2282.612997, 15, 180.433238,
81981722, 669.23, 714.985299, 2243.931556, 16, 180.593931, 80265921, 648.98, 718.799725,
2253.81652, 17, 181.03378, 80771547, 660.75, 719.30464, 2262.87062, 18, 183.055691, 81261955,
668.45, 728.711208, 2293.784478, 19, 180.887973, 80911994, 664.16, 719.01968, 2249.179266,
20, 180.813943, 80232499, 664.85, 721.411324, 2284.652261, 21, 180.385756, 80605689, 648.82,
715.639205, 2245.864785, 22, 180.444606, 80330647, 646.54, 716.915016, 2242.344091, 23, 180.71431,
80737374, 675.04, 718.832714, 2265.009509, 24, 180.668524, 81238595, 658.58, 719.155668,
2255.558149, 25, 180.481454, 80345081, 665.36, 717.399933, 2263.858354, 26, 181.532654,
81249440, 664.69, 720.445472, 2270.803738, 27, 180.709031, 80656222, 667.62, 720.197503,
2254.577266, 28, 181.788008, 80430417, 659.34, 722.947092, 2266.789663, 29, 180.267631,
80966830, 680.79, 717.274946, 2254.756044, 30, 181.218337, 81063718, 660.46, 719.292641,
2254.97179, 31, 181.825644, 80450839, 641.76, 723.430013, 2264.734585, 32, 180.503321, 80865988,
680.6, 718.113825, 2261.126508, 33, 181.179506, 81782679, 663.26, 718.818714, 2253.825399,
34, 182.404721, 80874295, 681.88, 724.456863, 2274.085854, 35, 180.679285, 79950377, 696.9,
725.942636, 2391.798143, 36, 180.311331, 80278492, 654.51, 717.365939, 2247.676829, 37,
181.924858, 80630340, 678.34, 722.615137, 2277.206623, 38, 182.235345, 80395053, 657.49,
723.116065, 2267.382674, 39, 180.581138, 80707569, 650.81, 715.812174, 2241.112811, 40,
180.255312, 80230787, 654.34, 712.459686, 2238.932363, 41, 180.430825, 80303558, 674.26,
717.554909, 2260.552094, 42, 182.713491, 81026742, 664.82, 723.136062, 2273.304849, 43,
180.667787, 81123555, 646.33, 720.56145, 2255.96479, 44, 181.29933, 80633352, 655.06, 722.545152,
2267.466679, 45, 180.69981, 80423923, 651.97, 720.521456, 2266.314221, 46, 180.963379, 80321818,
666.05, 719.914553, 2261.325868, 47, 180.400886, 80927362, 660.59, 716.602059, 2258.954976,
48, 180.415696, 80766134, 643.91, 718.038834, 2247.010966, 49, 181.75904, 81363295, 677.94,
724.362873, 2280.662367, 50, 181.07667, 81109019, 655.92, 719.259652, 2265.467021, 51, 181.529131,
80501688, 678.88, 723.304033, 2272.354133, 52, 181.032646, 80787881, 663.19, 719.145665,
2261.789334, 53, 181.78241, 80696334, 667.26, 721.840262, 2249.778509, 54, 181.784114, 80527776,
654.37, 723.531001, 2273.24765, 55, 180.258385, 80829579, 658.72, 718.865711, 2250.767518,
56, 180.224655, 80378818, 664.98, 716.456076, 2253.322893, 57, 181.080928, 80368464, 674.52,
720.215501, 2244.64257, 58, 180.499002, 80380180, 648.42, 717.849869, 2237.419292, 59, 180.465156,
80358888, 657.07, 716.04914, 2241.740393, 60, 181.267363, 80990378, 662.39, 722.909094,
2277.058788, 61, 180.855174, 80440724, 652.84, 718.467618, 2250.585334, 62, 180.613888, 80270814,
```

64, 180.729359, 81614898, 657.44, 719.963546, 2256.552945, 65, 181.693264, 80504070, 655.128, 723.022081, 2269.861376, 66, 182.387453, 80874296, 671.79, 727.868337, 2290.131526, 67, 182.641568, 81000190, 666.62, 725.311733, 2265.765748, 68, 180.408242, 80384436, 654.97, 716.394082, 2246.176717, 69, 180.50043, 80370121, 660.63, 717.081976, 2251.3445, 70, 180.314275, 80345215, 650.6, 715.708192, 2247.81268, 71, 180.40946, 80257696, 662.51, 718.742726, 2249.446348, 72, 180.480706, 80338131, 674.93, 715.993143, 2254.3009, 73, 180.351546, 81319554, 669.06, 717.563909, 2246.555165, 74, 182.185068, 80510723, 662.05, 726.882498, 2294.560762, 75, 181.050377, 81255283, 676.56, 720.206509, 2261.448971, 76, 181.284847, 80803267, 652.98, 719.70558, 2246.495795, 77, 181.899364, 80817635, 673.14, 722.509158, 2274.024187, 78, 181.1632, 80464233, 656.59, 722.296186, 2268.188293, 79, 182.306303, 81020663, 673.9, 724.157908, 2277.208026, 80, 180.737421, 81004315, 664.76, 718.59975, 2259.766497, 81, 180.769474, 80525069, 637.37, 719.664587, 2259.574753, 82, 180.525071, 80802789, 673.54, 720.259497, 2264.746946, 83, 180.395285, 80553202, 671.09, 718.417781, 2251.014061, 84, 184.756211, 80609250, 687.29, 734.63931, 2313.735932, 85, 180.42041, 80854674, 658.92, 717.500919, 2253.364023, 86, 180.526513, 80760351, 687.49, 720.31449, 2256.060244, 87, 182.019263, 81169002, 669.62, 724.327882, 2270.602457, 88, 181.050207, 80515085, 660.97, 719.636595, 2258.988494, 89, 182.15461, 81070869, 662.32, 722.750119, 2274.249861, 90, 180.647218, 80381001, 665.11, 721.512309, 2275.705292};

ss2048 = {1, 180.670493, 84068118, 590.35, 718.200808, 2270.345895, 2, 180.373626, 84641451, 598.23, 717.003993, 2357.850512, 3, 180.32103, 83894799, 544.09, 716.761031, 2220.56933, 4, 181.035053, 84104334, 537.93, 721.154361, 2241.816195, 5, 180.423765, 85285141, 537.28, 718.751725, 2239.623513, 6, 180.501278, 83946203, 552.66, 718.034838, 2228.650604, 7, 180.505279, 84031519, 537.83, 718.261802, 2236.537444, 8, 180.666334, 84725603, 544.91, 720.643436, 2233.047008, 9, 180.610769, 84809433, 541.2, 719.170667, 2233.254559, 10, 180.496245, 83760262, 544.19, 718.564754, 2243.495344, 11, 180.48963, 83973661, 550.24, 718.657745, 2228.042217, 12, 180.271806, 84319508, 544.23, 717.480919, 2232.600315, 13, 183.897054, 84308104, 554.7, 731.021862, 2311.08947, 14, 180.591717, 84761769, 551.91, 716.39609, 2217.092824, 15, 180.587998, 83949240, 530.69, 718.266795, 2229.805488, 16, 180.220937, 84259047, 550.79, 717.97384, 2243.213936, 17, 180.352575, 84500874, 558.4, 716.121126, 2229.678209, 18, 180.314325, 83885455, 561.27, 717.064987, 2223.717991, 19, 180.56337, 84088657, 517.22, 720.609441, 2232.955054, 20, 180.61446, 83964452, 572.33, 720.320488, 2247.416381, 21, 182.543394, 84029454, 524.87, 727.696376, 2260.68437, 22, 180.420452, 84111878, 547.11, 720.75142, 2233.470314, 23, 181.412112, 84867215, 562.18, 723.940937, 2251.776322, 24, 181.791536, 84060695, 531.33, 724.486858, 2240.094852, 25, 182.865567, 84041411, 525.81, 727.864337, 2271.634986, 26, 182.360544, 85635081, 535.66, 725.21274, 2254.394408, 27, 180.309633, 83807503, 541.44, 715.336248, 2222.246231, 28, 181.680407, 84808859, 549.47, 722.147213, 2254.855111, 29, 181.187124, 83658138, 553.79, 718.785719, 2240.036348, 30, 180.390909, 84299236, 558.29, 717.802871, 2231.522175, 31, 180.466954, 84461474, 561.38, 717.291947, 2232.168352, 32, 180.294102, 83881032, 525.65, 714.760338, 2220.980808, 33, 180.418724, 83890401, 539.17, 717.40493, 2236.75437, 34, 182.597286, 84264131, 555.83, 727.592381, 2257.93583, 35, 181.801385, 84675691, 565.1, 724.744814, 2258.649221, 36, 180.823096, 84136280, 562.5, 719.886554, 2242.427118, 37, 180.50522, 84627217, 528.67, 718.557758, 2236.363205, 38, 180.673101, 83865153, 545.17, 718.704739, 2235.846486, 39, 181.970873, 84038523, 548.77, 724.444858, 2255.211579, 40, 181.838359, 84614414, 533.1, 722.732122, 2241.870712, 41, 180.486088, 84206978, 542.43, 719.45062, 2248.558048, 42, 182.524343, 84884206, 552.39, 726.55154, 2261.429148, 43, 180.776978, 84456433, 538.11, 720.059532, 2241.26599, 44, 180.963913, 84755538, 542.52, 719.956545, 2233.193959, 45, 183.530741, 85561849, 550.88, 731.094855, 2274.52935, 46, 180.69616, 83997515, 573.46, 719.365631, 2234.828494, 47, 180.489986, 83500235, 795.25, 716.310089, 2230.680066, 48, 181.677741, 83930620, 543.55, 723.463008, 2255.478566, 49, 182.07414, 84490404, 583.71, 728.851196, 2270.045329, 50, 180.792824, 84422582, 547.41, 719.790571, 2230.593103, 51, 180.703274, 84549301, 555.63, 717.143964, 2233.371813, 52, 180.723654, 83725484, 551.7, 719.017686, 2227.106308, 53, 180.711647, 83685105, 536.39, 716.734032, 2230.332304, 54, 180.562286, 83895169, 535.03, 716.050136, 2229.134779, 55, 181.360642, 84584025, 545.54, 723.106062, 2249.121993, 56, 181.902422, 84132498, 544.12, 722.577141, 2242.436679, 57, 180.854107, 84984350, 567.39, 718.675738, 2230.992559, 58, 181.249345, 83791950, 560.06, 719.351637, 2238.190583, 59, 182.253847, 84820374, 592.82, 724.392869, 2280.408337, 60, 182.853205, 82830795, 670.46, 727.221453, 2450.343031, 61, 180.523547, 83356345, 712.16, 715.46023, 2402.083167, 62, 184.280039, 83512963, 771.26, 732.730611, 2470.673997, 63, 181.693067, 83924066, 589.18, 753.019523, 2906.918644, 64, 181.520418, 83865498, 573.54, 723.06807, 2242.530433, 65, 182.344888, 83981412, 558.09, 726.784508, 2270.441801, 66, 185.170334, 84657608, 591, 738.303753, 2307.428875, 67, 182.931815, 83587805, 582.56, 729.128149, 2279.956815, 68, 183.155708, 83971077, 561.43, 727.626381, 2263.099615, 69, 180.48389, 83731032, 576.62, 717.167971, 2235.231093, 70, 181.253942, 83829585, 563.51, 722.746116, 2248.812509, 71, 180.576187, 84192900, 575.3, 718.830716, 2236.608953, 72, 180.733868, 84367420, 567.01, 719.240652, 2241.7931, 73, 180.512719, 83350317, 539.42, 715.286251, 2227.405567, 74, 180.651249, 83822882, 543.19, 715.982155, 2225.121732, 75, 180.589051, 83550733, 577.41, 718.418781, 2260.940951, 76, 180.688569, 83522352, 564.09, 719.358626, 2243.675975, 77, 180.684587, 83635175, 559.5, 719.31664, 2239.909615, 78, 180.516067, 83535638, 564.4, 717.092974, 2225.661682, 79, 180.481876, 83387972, 579.6, 719.700581, 2238.532358, 80, 181.483102, 83723920, 560.73, 721.5603, 2249.322095, 81, 183.399369, 83800335, 551.6, 727.436405, 2265.263619, 82, 181.980817, 85144352, 561.33, 726.351573, 2253.703488, 83, 181.873525, 83630855, 567.29, 724.197904, 2254.186829, 84, 182.207331, 84328011, 558.53, 728.195286, 2267.311053, 85, 180.413211, 84051672, 567.47, 717.187966, 2227.403915, 86, 180.919276, 84591856, 555.59, 721.656284, 2248.592958, 87, 180.762429, 84064408, 548.85, 719.097682, 2226.724287, 88, 181.256613, 83576165, 558.42, 721.111365, 2243.209287, 89, 181.8704, 83582540, 556.62, 722.344187, 2244.681682, 90, 180.650170, 85022325, 565.55, 720.131526, 2241.870712};

ss8192 = {1, 180.443782, 83 573 198, 472.42, 719.890548, 2252.866921, 2, 180.538482, 83 805 895, 450.82, 715.556223, 2225.502126, 3, 180.418896, 83 609 982, 451.92, 716.982991, 2234.628367, 4, 181.522059, 84 215 310, 454.22, 724.099915, 2261.259502, 5, 181.792504, 84 562 525, 480.79, 724.49086, 2249.694105, 6, 182.041045, 83 882 991, 453.96, 725.76866, 2271.116145, 7, 181.253696, 84 474 279, 460.76, 722.81211, 2250.877371, 8, 181.093033, 84 220 755, 471.92, 720.760422, 2257.331153, 9, 182.205761, 84 683 211, 469.58, 727.657374, 2272.333074, 10, 181.561697, 84 831 286, 484.07, 720.768423, 2255.671482, 11, 183.403127, 83 782 219, 455.34, 730.253978, 2281.73003, 12, 181.973115, 83 567 614, 464.78, 724.438859, 2271.264626, 13, 180.618458, 84 770 777, 450.36, 719.971543, 2242.561082, 14, 180.637587, 83 711 742, 460.17, 720.607442, 2252.886592, 15, 180.318941, 83 669 617, 461.97, 720.429477, 2249.516546, 16, 180.49561, 84 856 040, 459.48, 716.582058, 2241.048394, 17, 182.856464, 85 039 119, 473.41, 729.262128, 2280.5888, 18, 183.816034, 84 306 706, 486.07, 731.068855, 2303.344625, 19, 183.708645, 83 846 431, 463.23, 732.080707, 2289.154645, 20, 181.844555, 84 442 880, 478.92, 723.747963, 2251.21474, 21, 180.677098, 83 575 181, 445.82, 719.132671, 2244.371814, 22, 180.823296, 84 003 391, 440.9, 719.385636, 2237.485878, 23, 181.385144, 84 796 781, 457.19, 720.211502, 2245.955804, 24, 180.833268, 83 943 545, 471.56, 720.421474, 2248.450668, 25, 180.901224, 84 572 897, 487.94, 720.318495, 83 242 523, 26, 181.552319, 22 287 557, 475.75, 726.129, 2254.82523, 27, 181.923255, 84 657 657, 483.52, 727.657374, 2272.333074, 28, 181.561697, 84 831 286, 484.07, 720.768423, 2255.671482, 29, 183.403127, 83 782 219, 455.34, 730.253978, 2281.73003, 30, 181.973115, 83 567 614, 464.78, 724.438859, 2271.264626, 31, 180.618458, 84 770 777, 450.36, 719.971543, 2242.561082, 32, 180.637587, 83 711 742, 460.17, 720.607442, 2252.886592, 33, 180.318941, 83 669 617, 461.97, 720.429477, 2249.516546, 34, 180.49561, 84 856 040, 459.48, 716.582058, 2241.048394, 35, 182.856464, 85 039 119, 473.41, 729.262128, 2280.5888, 36, 183.816034, 84 306 706, 486.07, 731.068855, 2303.344625, 37, 183.708645, 83 846 431, 463.23, 732.080707, 2289.154645, 38, 181.844555, 84 442 880, 478.92, 723.747963, 2251.21474, 39, 180.677098, 83 575 181, 445.82, 719.132671, 2244.371814, 40, 180.823296, 84 003 391, 440.9, 719.385636, 2237.485878, 41, 181.385144, 84 796 781, 457.19, 720.211502, 2245.955804, 42, 180.833268, 83 943 545, 471.56, 720.421474, 2248.450668, 43, 180.901224, 84 572 897, 487.94, 720.318495, 83 242 523, 44, 181.552319, 22 287 557, 475.75, 726.129, 2254.82523, 45, 181.923255, 84 657 657, 483.52, 727.657374, 2272.333074, 46, 181.561697, 84 831 286, 484.07, 720.768423, 2255.671482, 47, 183.403127, 83 782 219, 455.34, 730.253978, 2281.73003, 48, 181.973115, 83 567 614, 464.78, 724.438859, 2271.264626, 49, 180.618458, 84 770 777, 450.36, 719.971543, 2242.561082, 50, 180.637587, 83 711 742, 460.17, 720.607442, 2252.886592, 51, 180.318941, 83 669 617, 461.97, 720.429477, 2249.516546, 52, 180.49561, 84 856 040, 459.48, 716.582058, 2241.048394, 53, 182.856464, 85 039 119, 473.41, 729.262128, 2280.5888, 54, 183.816034, 84 306 706, 486.07, 731.068855, 2303.344625, 55, 183.708645, 83 846 431, 463.23, 732.080707, 2289.154645, 56, 181.844555, 84 442 880, 478.92, 723.747963, 2251.21474, 57, 180.677098, 83 575 181, 445.82, 719.132671, 2244.371814, 58, 180.823296, 84 003 391, 440.9, 719.385636, 2237.485878, 59, 181.385144, 84 796 781, 457.19, 720.211502, 2245.955804, 60, 180.833268, 83 943 545, 471.56, 720.421474, 2248.450668, 61, 180.901224, 84 572 897, 487.94, 720.318495, 83 242 523, 62, 181.552319, 22 287 557, 475.75, 726.129, 2254.82523, 63, 181.923255, 84 657 657, 483.52, 727.657374, 2272.333074, 64, 181.561697, 84 831 286, 484.07, 720.768423, 2255.671482, 65, 183.403127, 83 782 219, 455.34, 730.253978, 2281.73003, 66, 181.973115, 83 567 614, 464.78, 724.438859, 2271.264626, 67, 180.618458, 84 770 777, 450.36, 719.971543, 2242.561082, 68, 180.637587, 83 711 742, 460.17, 720.607442

29, 182.593187, 84 383 396, 490.9, 727.905339, 2279.264212, 30, 181.407053, 83 904 730, 477.58, 724.027921, 2259.655646, 31, 180.340899, 84 891 140, 476.17, 715.744184, 2253.050643, 32, 180.821783, 83 755 948, 452.62, 720.102522, 2247.615427, 33, 181.268671, 84 750 866, 462.47, 721.14637, 2258.018607, 34, 181.405192, 83 858 157, 476, 724.417862, 2278.260321, 35, 180.988866, 84 748 114, 463.62, 720.390475, 2253.53767, 36, 180.552143, 83 957 966, 446.68, 719.800565, 2228.161635, 37, 180.790724, 84 026 308, 474.11, 720.48947, 2267.694443, 38, 181.304933, 83 966 979, 465.48, 720.366475, 2247.145261, 39, 181.994626, 84 325 776, 468.16, 722.60914, 2255.140915, 40, 180.55589, 84 167 553, 485.31, 719.472613, 2252.122588, 41, 181.718677, 84 958 089, 450.79, 723.920937, 2257.357576, 42, 180.64837, 85 255 296, 489.64, 714.220413, 2234.246162, 43, 180.539746, 83 715 741, 467.92, 718.925702, 2250.013227, 44, 180.958981, 82 929 087, 481.09, 726.882498, 2491.429596, 45, 180.353289, 83 575 547, 462.32, 716.213109, 2239.913753, 46, 180.680464, 84 147 394, 508.86, 718.745731, 2243.076927, 47, 184.016646, 86 452 375, 481.53, 732.865582, 2292.397966, 48, 182.751857, 84 790 370, 454.82, 727.910334, 2262.280703, 49, 180.86567, 84 379 012, 463.32, 720.637439, 2247.543524, 50, 180.687835, 83 692 524, 457.73, 717.041994, 2237.230355, 51, 180.613981, 83 752 606, 462.59, 719.708581, 2259.548728, 52, 180.423803, 83 808 530, 462.13, 718.605746, 2235.919, 53, 183.688402, 83 948 672, 474.3, 733.047554, 2303.809943, 54, 182.393027, 84 556 286, 465.68, 726.858492, 2268.948485, 55, 180.675415, 83 890 822, 470.73, 723.063075, 2261.787918, 56, 182.367624, 84 309 392, 460.7, 725.396717, 2268.162595, 57, 181.899311, 84 410 425, 467.9, 723.981938, 2259.043231, 58, 183.368589, 84 651 734, 481.91, 728.820197, 2280.665848, 59, 180.700908, 84 005 443, 444.86, 719.928549, 2243.353469, 60, 182.85822, 85 550 922, 465.63, 727.061468, 2268.900242, 61, 182.379904, 84 527 556, 462.14, 724.351879, 2256.180509, 62, 183.401636, 84 247 535, 470.53, 730.542931, 2277.195578, 63, 181.714025, 83 877 146, 481.23, 722.943091, 2268.672645, 64, 183.793536, 83 892 773, 479.98, 732.360657, 2295.249743, 65, 181.023319, 84 114 813, 461.38, 722.438172, 2260.272227, 66, 180.568441, 84 676 603, 487.01, 718.116826, 2247.292308, 67, 180.505907, 83 682 463, 477.15, 719.895551, 2243.691278, 68, 180.888744, 83 500 434, 488.11, 720.391474, 2254.677328, 69, 180.556586, 83 745 620, 457.06, 717.868859, 2245.885961, 70, 184.844165, 83 665 109, 471.13, 737.583866, 2314.390165, 71, 182.390927, 83 834 672, 473.04, 726.52255, 2268.697948, 72, 180.509811, 84 084 027, 469.94, 715.887162, 2238.337387, 73, 180.376337, 83 470 946, 454.97, 717.6059, 2242.037534, 74, 180.866471, 83 733 055, 477.41, 721.00438, 2246.451313, 75, 180.543335, 83 864 870, 462.32, 720.649438, 2263.205507, 76, 180.714309, 85 137 564, 469.37, 720.455477, 2258.468815, 77, 180.718048, 83 736 079, 459.02, 721.5193, 2265.087913, 78, 180.26687, 84 231 785, 470.7, 719.381632, 2249.646307, 79, 181.444991, 84 378 948, 452.9, 724.793804, 2273.918272, 80, 182.676355, 84 715 820, 458.57, 726.893491, 2266.581513, 81, 183.72009, 84 824 711, 461.42, 731.650765, 2294.650742, 82, 181.864416, 85 438 268, 469.7, 724.309884, 2254.368804, 83, 182.062629, 84 552 744, 478.45, 724.596835, 2267.317402, 84, 181.749335, 83 494 307, 447.24, 723.003087, 2253.499157, 85, 183.806769, 84 180 149, 428.35, 733.358514, 2283.48773, 86, 181.692507, 84 010 948, 476.47, 726.643531, 2273.157771, 87, 181.872797, 84 332 429, 460.54, 724.324877, 2264.166512, 88, 187.614585, 84 545 101, 484.03, 743.296, 2332.082538, 89, 180.576782, 85 406 163, 467.72, 718.532758, 2244.330955, 90, 180.553742, 83 650 885, 457.79, 718.195812, 2250.301638};

ssl16384 = {1, 180.953645, 83 470 361, 380.65, 719.775577, 2259.480174, 2, 186.426605, 85 020 624, 402.27, 741.732232, 2340.809435, 3, 182.315004, 83 671 880, 390, 727.519397, 2301.715394, 4, 180.738147, 84 212 807, 381.12, 721.18736, 2269.765994, 5, 183.761411, 83 727 194, 390.55, 734.195376, 2316.33754, 6, 183.489898, 83 699 399, 385.91, 731.633767, 2310.159767, 7, 182.348803, 83 423 284, 404.93, 727.010468, 2296.115189, 8, 182.202339, 84 250 006, 383.42, 725.041769, 2277.282037, 9, 184.746653, 83 763 810, 402.86, 735.087246, 2332.478285, 10, 182.055514, 85 865 379, 400.6, 724.74282, 2281.34664, 11, 180.532821, 83 812 749, 394.78, 721.359333, 2279.092941, 12, 180.409979, 83 145 551, 382.81, 719.929543, 2270.154804, 13, 180.470685, 83 191 499, 389.39, 720.083529, 2272.06255, 14, 180.704518, 83 354 467, 401.11, 720.601451, 2270.109596, 15, 180.555058, 84 087 838, 386.27, 719.294643, 2272.071156, 16, 180.806183, 83 142 430, 384.94, 720.714425, 2269.543478, 17, 180.603517, 83 122 769, 374.93, 720.078532, 2271.459927, 18, 182.680297, 84 569 309, 428.77, 729.879039, 2453.477135, 19, 180.234802, 84 196 255, 439.77, 731.877735, 2534.623032, 20, 180.432128, 83 037 148, 418.33, 717.433932, 2265.661121, 21, 184.13468, 83 909 220, 395.98, 733.503485, 2319.075676, 22, 186.08837, 85 100 630, 399.32, 740.424427, 2336.561428, 23, 182.347956, 83 603 615, 381.19, 728.077312, 2292.479044, 24, 180.442707, 84 442 636, 398.47, 719.325641, 2256.809542, 25, 182.53302, 84 155 230, 372.95, 729.271123, 2290.632092, 26, 181.12545, 84 908 512, 404.11, 720.65044, 2271.980504, 27, 182.862084, 84 394 760, 410.12, 727.622377, 2302.946293, 28, 180.430228, 84 890 033, 377.24, 717.843858, 2252.22843, 29, 180.295424, 83 464 375, 364.99, 721.563305, 2265.187557, 30, 180.223411, 83 262 761, 386.93, 717.554907, 2268.968798, 31, 180.595459, 83 071 822, 367.77, 720.688435, 2272.264127, 32, 180.465603, 83 042 571, 374.22, 720.788414, 2268.216543, 33, 180.461698, 83 104 451, 383.27, 718.164824, 2264.090513, 34, 181.984514, 83 929 931, 407.43, 725.012771, 2278.268857, 35, 181.450081, 83 745 065, 400.23, 720.8944, 2271.626027, 36, 181.543443, 83 049 463, 392.23, 723.816959, 2275.364705, 37, 182.084224, 83 504 341, 389.25, 724.475855, 2273.442378, 38, 182.179003, 84 183 753, 376.07, 729.108155, 2288.452062, 39, 181.56577, 83 414 213, 376.15, 722.016227, 2273.773552, 40, 180.520559, 83 725 102, 367.95, 718.407782, 2260.880257, 41, 180.302341, 83 744 170, 401.44, 718.207805, 2261.130905, 42, 180.69098, 84 081 363, 399.84, 720.564452, 2263.341085, 43, 181.603401, 84 492 728, 399.75, 724.594836, 2278.328191, 44, 183.252715, 83 661 056, 386.19, 729.982018, 2298.387713, 45, 181.688666, 83 372 864, 392.84, 721.726272, 2253.725255, 46, 184.157977, 84 792 895, 406.43, 733.069553, 2310.280143, 47, 181.79921, 84 792 040, 401.33, 723.124064, 2275.19847, 48, 184.965828, 83 424 009, 399.33, 738.533715, 2325.436615, 49, 180.778851, 84 949 236, 385.01, 720.030529, 2263.960426, 50, 183.016715, 83 514 439, 389.02, 727.005479, 2294.955241, 51, 181.521473, 83 602 672, 395.85, 721.698276, 2274.937952, 52, 181.382571, 83 554 600, 380.71, 721.882350, 2270.875001, 53, 181.848711, 83 870 300, 367.85, 720.265000, 2275.440000, 54, 180.400000, 83 000 000, 300.00, 700.000000, 2200.000000, 55, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 56, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 57, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 58, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 59, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 60, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 61, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 62, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 63, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 64, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 65, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 66, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 67, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 68, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 69, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 70, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 71, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 72, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 73, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 74, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 75, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 76, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 77, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 78, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 79, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 80, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 81, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 82, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 83, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 84, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 85, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 86, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 87, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 88, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 89, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 90, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 91, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 92, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 93, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 94, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 95, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 96, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 97, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 98, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 99, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 100, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 101, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 102, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 103, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 104, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 105, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 106, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 107, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 108, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 109, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 110, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 111, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 112, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 113, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 114, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 115, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 116, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 117, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 118, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 119, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 120, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 121, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 122, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 123, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 124, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 125, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 126, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 127, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 128, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 129, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 130, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 131, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 132, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 133, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 134, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 135, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 136, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 137, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 138, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 139, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 140, 180.000000, 83 000 000, 300.00, 700.000000, 2200.000000, 141, 180.000000, 8



8931007, 719.977544, 2262.009545, 56, 180.670566, 83 469 408, 382.55, 722.022233, 2274.203967, 57, 180.525231, 84 956 940, 416.48, 720.702437, 2269.818236, 58, 180.569001, 83 343 932, 392.29, 721.969241, 2275.102701, 59, 180.818799, 83 308 560, 420.41, 721.733281, 2278.385695, 60, 180.573712, 84 980 176, 401.07, 718.680735, 2277.328855, 61, 180.606785, 84 224 204, 392.25, 718.696741, 2260.504628, 62, 180.358386, 83 164 528, 388.64, 717.645898, 2261.189334, 63, 180.595804, 83 137 052, 390.28, 720.020534, 2254.410532, 64, 180.469438, 83 303 112, 369.16, 716.512071, 2252.933757, 65, 182.1805, 83 253 165, 387.67, 725.510706, 2289.783954, 66, 181.89181, 84 867 414, 399.24, 724.680832, 2283.076769, 67, 183.613854, 83 603 321, 386.31, 730.282972, 2294.295636, 68, 182.551428, 83 176 348, 401.19, 729.973019, 2295.526612, 69, 180.257517, 83 729 785, 399.13, 718.639746, 2266.608968, 70, 182.493877, 83 668 926, 409.16, 725.532695, 2287.862351, 71, 181.367754, 83 966 913, 376.86, 722.891097, 2289.10394, 72, 182.735691, 85 932 264, 404.29, 727.625377, 2290.180239, 73, 185.963691, 84 109 472, 386.28, 741.030343, 2342.676138, 74, 181.622432, 85 374 849, 399.07, 724.056921, 2273.954351, 75, 183.195858, 84 774 351, 515.77, 727.587388, 2295.123907, 76, 183.486335, 84 212 575, 377.65, 731.252823, 2295.043119, 77, 181.815282, 82 905 611, 416.19, 723.581991, 2288.118635, 78, 184.107622, 83 308 961, 395.79, 732.087698, 2304.020296, 79, 180.701647, 82 875 017, 387.75, 721.040378, 2261.70219, 80, 180.700975, 82 949 625, 391.79, 717.563908, 2266.016983, 81, 180.463036, 82 962 295, 386.39, 719.620594, 2265.7017, 82, 180.447392, 82 911 524, 406.37, 716.171121, 2258.585797, 83, 182.793, 83 104 345, 400.64, 728.312271, 2305.272539, 84, 183.389713, 84 273 300, 392.36, 729.031163, 2300.998827, 85, 180.586179, 83 654 002, 401.06, 718.887706, 2266.15853, 86, 180.676902, 82 957 280, 381.18, 721.441312, 2267.351109, 87, 180.461958, 82 850 968, 401.86, 717.944856, 2266.453068, 88, 183.085924, 84 042 493, 415.45, 729.456093, 2316.388468, 89, 181.237922, 83 845 578, 394.28, 720.080527, 2268.701405, 90, 180.551184, 84 386 400, 404.59, 717.44893, 2263.911133};

ss32768 = {1, 180.569869, 83 360 221, 307.32, 720.805417, 2267.887399, 2, 182.227002, 84 343 746, 301.45, 726.750507, 2272.073482, 3, 181.987718, 85 575 309, 321.26, 723.756973, 2280.593978, 4, 181.658226, 84 484 802, 315.71, 723.219051, 2256.686288, 5, 180.586187, 85 462 888, 295.85, 717.152966, 2244.772683, 6, 182.773148, 85 503 169, 313.49, 728.794198, 2263.016746, 7, 187.662372, 83 638 910, 311.77, 749.173101, 2358.757703, 8, 182.305362, 85 586 988, 310.11, 726.721518, 2281.242177, 9, 185.086702, 83 312 853, 340.75, 736.10609, 2474.715978, 10, 180.881734, 83 481 126, 340.81, 733.584477, 2522.836063, 11, 183.1089, 86 212 275, 346.74, 731.344814, 2291.057769, 12, 180.388539, 84 138 002, 332.65, 717.290952, 2263.639503, 13, 180.269646, 84 050 300, 313.51, 719.041679, 2260.4275, 14, 180.435415, 83 510 955, 320.65, 714.969301, 2262.102862, 15, 180.592446, 83 257 467, 306.4, 720.685437, 2250.265681, 16, 183.953501, 85 275 444, 319.08, 730.622919, 2292.241723, 17, 180.690136, 84 749 828, 322.48, 718.066827, 2256.860819, 18, 180.250711, 83 669 694, 317.36, 717.035987, 2245.593275, 19, 184.994001, 86 371 142, 310.92, 737.624857, 2317.653304, 20, 182.984634, 85 141 784, 316.79, 730.93687, 2287.299778, 21, 184.841999, 86 369 339, 321.17, 736.489028, 2325.581383, 22, 183.50748, 85 018 984, 298.09, 731.303818, 2281.955032, 23, 190.072778, 83 544 077, 315.36, 756.369019, 2388.400148, 24, 180.434884, 85 319 509, 305.84, 721.580299, 2246.236376, 25, 181.988611, 84 157 953, 319.41, 724.445856, 2266.834472, 26, 187.887893, 85 524 605, 323.01, 749.490057, 2362.953073, 27, 184.272297, 88 030 722, 306.34, 732.422644, 2283.46806, 28, 181.511801, 84 101 040, 311.88, 724.730815, 2291.868066, 29, 185.662589, 85 340 326, 311.08, 739.557561, 2315.774201, 30, 185.904663, 85 084 176, 308.66, 740.578414, 2337.653878, 31, 182.693608, 86 009 886, 305.94, 727.148448, 2274.930672, 32, 180.4363, 83 801 411, 293.7, 719.002689, 2250.754247, 33, 185.543405, 85 304 281, 304.28, 741.005343, 2318.313549, 34, 181.549513, 85 842 510, 298.99, 724.926786, 2268.479687, 35, 184.884289, 83 792 456, 303.64, 736.223071, 2316.618908, 36, 180.409835, 84 926 110, 308.13, 716.142126, 2243.687435, 37, 180.276097, 83 488 905, 327.57, 718.977688, 2263.434078, 38, 182.048616, 83 570 775, 305.68, 723.790961, 2268.248721, 39, 184.585056, 84 084 641, 309.47, 736.016102, 2310.607052, 40, 181.289152, 84 560 963, 336.76, 721.95724, 2270.788932, 41, 183.92803, 83 820 217, 385.53, 736.46904, 2517.793004, 42, 184.959755, 83 598 205, 307.48, 738.686698, 2314.483725, 43, 183.262684, 84 629 567, 316.9, 729.951029, 2281.579854, 44, 181.331268, 83 499 091, 318.22, 723.131057, 2253.804488, 45, 182.598446, 84 564 431, 314.02, 729.84304, 2315.089037, 46, 185.149163, 84 806 076, 326.9, 738.14078, 2323.655024, 47, 182.597906, 84 790 095, 297.06, 725.43172, 2262.986724, 48, 180.825069, 83 990 519, 320.22, 719.082666, 2257.335479, 49, 182.647488, 84 733 422, 310.52, 728.284281, 2287.070073, 50, 180.577861, 83 516 961, 296.44, 719.801571, 2248.143801, 51, 185.625999, 85 489 242, 302.58, 746.115569, 2326.315295, 52, 186.374626, 83 864 701, 308.68, 743.468973, 2334.829227, 53, 180.55446, 84 066 818, 298.48, 719.561604, 2254.601766, 54, 180.785226, 84 347 743, 307.92, 721.129361, 2262.016741, 55, 180.738978, 84 531 798, 315.82, 721.011385, 2276.074448, 56, 181.171309, 84 111 931, 318.5, 721.824265, 2240.800344, 57, 180.335273, 83 631 599, 305.99, 716.172121, 2234.975043, 58, 180.737778, 83 363 406, 314.17, 720.613445, 2269.708893, 59, 185.1209, 85 577 355, 304.69, 740.312456, 2303.796329, 60, 181.161272, 83 935 517, 296.56, 724.780811, 2263.418365, 61, 180.721202, 85 321 650, 295.06, 720.622438, 2254.421013, 62, 180.602415, 85 023 762, 294.66, 719.778573, 2249.881011, 63, 181.311256, 85 533 581, 325.07, 722.223192, 2273.625164, 64, 180.619685, 84 928 319, 322.77, 720.513461, 2245.112311, 65, 180.449138, 83 415 902, 305.29, 719.246648, 2263.207407, 66, 180.490112, 83 997 840, 309.55, 720.456467, 2267.615624, 67, 183.755856, 84 058 117, 321.34, 731.954718, 2291.23878, 68, 180.338924, 83 818 977, 322.69, 718.954699, 2260.410088, 69, 180.460367, 84 088 360, 319.33, 720.235497, 2270.989964, 70, 181.080651, 83 930 219, 316.65, 720.851411, 2276.020692, 71, 180.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084, 313.86, 755.792096, 2387.867394, 73, 180.66762, 83 265 606, 299.81, 721.094367, 2257.57555, 74, 181.553707, 85 424 922, 318.8, 724.916796, 2268.362748, 75, 180.46591, 84 073 985, 337.5, 717.486919, 2262.855766, 76, 181.703432, 83 467 828, 309.42, 720.979385, 2268.449045, 77, 180.219127, 83 545 997, 318.47, 719.092676, 2253.415108, 78, 185.668105, 85 728 119, 320.42, 720.421550, 2254.221105, 79, 181.612692, 83 599 121, 298.57, 721.766268, 2249.079069, 72, 189.365464, 84 006 084,

```
6 716.362092, 2242.510759, 82, 184.781191, 86 061 030, 318.55, 737.698852, 2327.852006, 83, E1 CBC_Analysis_1d.nb
183.527645, 87 477 015, 314.29, 731.385809, 2296.093181, 84, 184.082349, 85 799 304, 340.46,
732.117699, 2291.987069, 85, 182.817882, 85 113 510, 317.22, 728.26228, 2290.655094, 86, 180.657898,
84 113 999, 311.06, 721.746273, 2263.252561, 87, 180.509809, 84 047 412, 303.63, 720.226499,
2254.124667, 88, 180.424072, 83 524 891, 293.32, 720.797421, 2252.298694, 89, 180.619329, 83 449 330,
315.69, 719.05868, 2262.201645, 90, 188.387692, 86 420 600, 338.32, 749.717023, 2369.01914};
```

## Data Loading

All the data sets are contained in the above section.

In[7]:=

```
ssNum = 6;
sampleNum = 90;
ss[1] = ss1024; latches[1] = 1024;
ss[2] = ss2048; latches[2] = 2048;
ss[3] = ss4096; latches[3] = 4096;
ss[4] = ss8192; latches[4] = 8192;
ss[5] = ss16384; latches[5] = 16384;
ss[6] = ss32768; latches[6] = 32768;

ncols = 6;
sampleCol = 1;
elapsedTCol = 2;
lioCol = 3;
waitTCol = 4;
cpuTCol = 5;
dbtimeCol = 6;

Do[
  ssLio[ssidx] = {}; ssL[ssidx] = {}; ssSt[ssidx] = {}; ssQt[ssidx] = {}; ssRt[ssidx] = {};
  theSS = ss[ssidx];
  Table[
    elapsedT = theSS[[ncols sampleidx + elapsedTCol]];
    lioTot = theSS[[ncols sampleidx + lioCol]];
    cpuSecTot = theSS[[ncols sampleidx + cpuTCol]];
    waitSecTot = theSS[[ncols sampleidx + waitTCol]];

    λ = lioTot / (1000 elapsedT);
    St = (cpuSecTot 1000) / lioTot;
    Qt = (waitSecTot 1000) / lioTot;
    Rt = St + Qt;

    AppendTo[ssLio[ssidx], lioTot];
    AppendTo[ssL[ssidx], λ];
    AppendTo[ssSt[ssidx], St];
    AppendTo[ssQt[ssidx], Qt];
    AppendTo[ssRt[ssidx], Rt];

    , {sampleidx, 0, sampleNum - 1}
  ];
  , {ssidx, ssNum}
];
Length[ssLio[6]]
Take[ssLio[6], 5]
Mean[ssL[6]]
```

Out[23]=

```
90
```

Out[24]=

```
{83 360 221, 84 343 746, 85 575 309, 84 484 802, 85 462 888}
```

Out[25]=

```
463.521
```

In this section I calculate the basic statistics, such as the mean and median. My objective is to ensure the data has been collected and entered correctly and also to compare the two datasets to see if they appear to be different.

In[26]:=

```
myData = Table[
  {
    latches[ssidx], Mean[ssL[ssidx]], Mean[ssSt[ssidx]], Mean[ssQt[ssidx]], Mean[ssRt[ssidx]],
    Length[ssLio[ssidx]], N[StandardDeviation[ssL[ssidx]]], N[StandardDeviation[ssSt[ssidx]]],
    N[StandardDeviation[ssQt[ssidx]]], N[StandardDeviation[ssRt[ssidx]]]
  }, {ssidx, 1, ssNum}
];
toGrid = Prepend[myData, {"CBC\nlatches", "Avg L\n(lio/ms)", "Avg St\n(ms/lio)",
  "Avg Qt\n(ms/lio)", "Avg Rt\n(ms/lio)", "Samples", "Stdev L\n(lio/ms)",
  "Stdev St\n(ms/lio)", "Stdev Qt\n(ms/lio)", "Stdev Rt\n(ms/lio)"}];
Grid[
  toGrid,
  Frame →
  All]
```

CBC latches	Avg L (lio/ms)	Avg St (ms/lio)	Avg Qt (ms/lio)	Avg Rt (ms/lio)	Samples	Stdev L (lio/ms)	Stdev St (ms/lio)	Stdev Qt (ms/lio)	Stdev Rt (ms/lio)
1024	445.565	0.008927\35	0.008221\62	0.017149	90	2.76918	0.000061\5631	0.000139\055	0.000171\181
2048	464.279	0.0085746	0.006688\83	0.0152634	90	3.60708	0.000079\4641	0.000523\737	0.000557\329
4096	464.541	0.008569\27	0.006276\16	0.0148454	90	3.0744	0.000065\6271	0.000237\176	0.000254\089
8192	463.743	0.008587\73	0.005547\69	0.0141354	90	3.89317	0.000078\4617	0.000152\35	0.000179\97
16384	461.291	0.008639\03	0.004707\89	0.0133469	90	4.49303	0.000084\6721	0.000213\799	0.000231\229
32768	463.521	0.008602\26	0.0037164	0.0123187	90	6.61111	0.000129\064	0.000172\751	0.000243\412

Out[28]=

In this section I calculate the key parameters for understanding the impact of changing the number of latches and chains. The columns heading are queuing theory centric, not Oracle-ease centric.

In[29]:=

```
myData = Table[
  {
    latches[ssidx],
    Mean[ssL[ssidx]], 100 * (Mean[ssL[ssidx]] - Mean[ssL[1]]) / Mean[ssL[1]],
    Mean[ssSt[ssidx]], 100 * (Mean[ssSt[ssidx]] - Mean[ssSt[1]]) / Mean[ssSt[1]],
    Mean[ssQt[ssidx]], 100 * (Mean[ssQt[ssidx]] - Mean[ssQt[1]]) / Mean[ssQt[1]],
    Mean[ssRt[ssidx]], 100 * (Mean[ssRt[ssidx]] - Mean[ssRt[1]]) / Mean[ssRt[1]],
    Length[ssLio[ssidx]]
  }, {ssidx, 1, ssNum}
];
toGrid = Prepend[myData, {"CBC\nlatches", "Avg L\n(lio/ms)", "%\nChange", "Avg St\n(ms/lio)",
  "%\nChange", "Avg Qt\n(ms/lio)", "%\nChange", "Avg Rt\n(ms/lio)", "%\nChange", "Samples"}];
Grid[
  toGrid,
  Frame →
  All]
```

CBC latches	Avg L (lio/ms)	% Change	Avg St (ms/lio)	% Change	Avg Qt (ms/lio)	% Change	Avg Rt (ms/lio)	% Change	Samples
1024	445.565	0.	0.00892735	0.	0.00822162	0.	0.017149	0.	90
2048	464.279	4.20005	0.0085746	-3.95135	0.00668883	-18.6434	0.0152634	-10.9951	90
4096	464.541	4.2588	0.00856927	-4.01106	0.00627616	-23.6628	0.0148454	-13.4326	90
8192	463.743	4.07982	0.00858773	-3.80429	0.00554769	-32.5232	0.0141354	-17.5728	90
16384	461.291	3.52953	0.00863903	-3.22963	0.00470789	-42.7377	0.0133469	-22.1707	90
32768	463.521	4.03003	0.00860226	-3.64153	0.0037164	-54.7973	0.0123187	-28.1668	90

Out[31]=

## Sample Set Normality Tests

Before we can perform a standard t-test hypothesis tests on our data, we need to ensure it is normally distributed...because that is one of the underlying assumptions and requirements for properly performing a t-test.

### Statistical and visual normality test

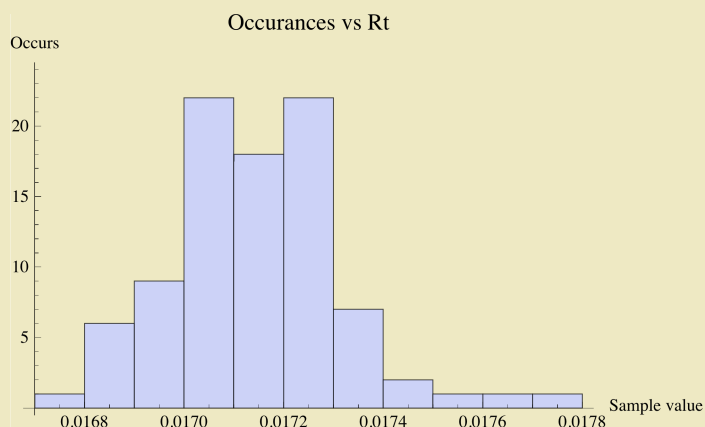
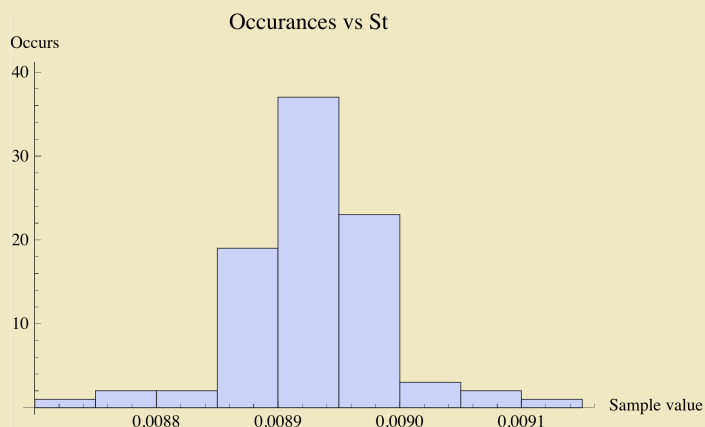
Our alpha will be 0.05, so if the distribution fit test results in a value greater than 0.05 then we can assume the data set is indeed normally distributed.

The first test is just to double check to make sure my thinking is correct. Since I creating a normal distribution based on a mean and standard deviation (just happens to be based on the my sample set data), I would expect a p-value (the result) to greatly exceed 0.05. Notice that the more samples I have created (the final number), the closer the p-value approaches 1.0.

In[49]:=

```
check = DistributionFitTest[
  RandomVariate[NormalDistribution[Mean[ssSt[1]], StandardDeviation[ssSt[1]], 10 000]];
Do[
  pValueSt = DistributionFitTest[ssSt[i]];
  pValueRt = DistributionFitTest[ssRt[i]];
  Print["Sample set ", i, " with ",
    Length[ssSt[i]], " values. P-values: St=", pValueSt, " Rt=", pValueRt];
  st = Histogram[ssSt[i], PlotLabel -> "Occurrences vs St", AxesLabel -> {"Sample value", "Occurs"}];
  rt = Histogram[ssRt[i], PlotLabel -> "Occurrences vs Rt", AxesLabel -> {"Sample value", "Occurs"}];
  Print[st];
  Print[rt];
  Print["-----"];
  , {i, 1, ssNum}
];
Print["This number should be much greater than 0.05: ",
  check, " If not try again by re-evaluating."];
```

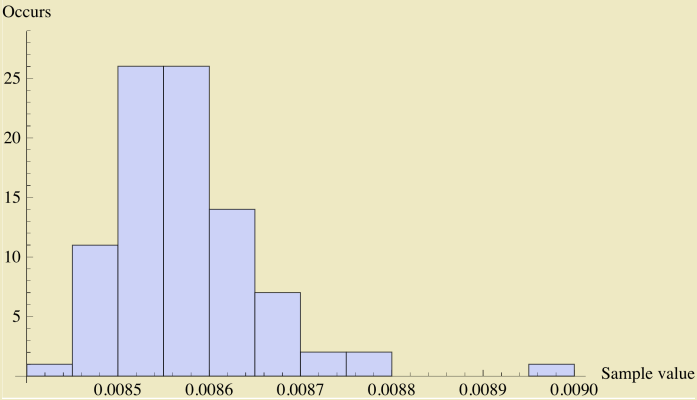
Sample set 1 with 90 values. P-values: St=0.0726837 Rt=0.455729



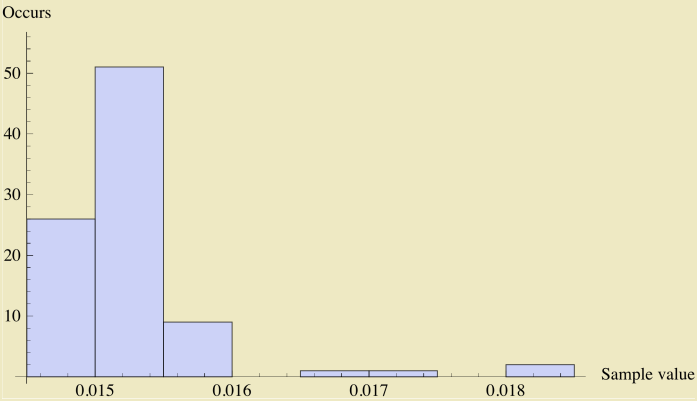
Sample set 2 with 90 values. P-values: St=0.00024229 Rt=0.



Occurances vs St

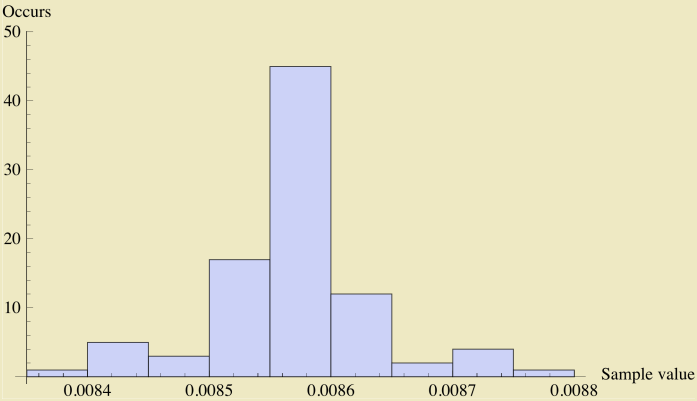


Occurances vs Rt

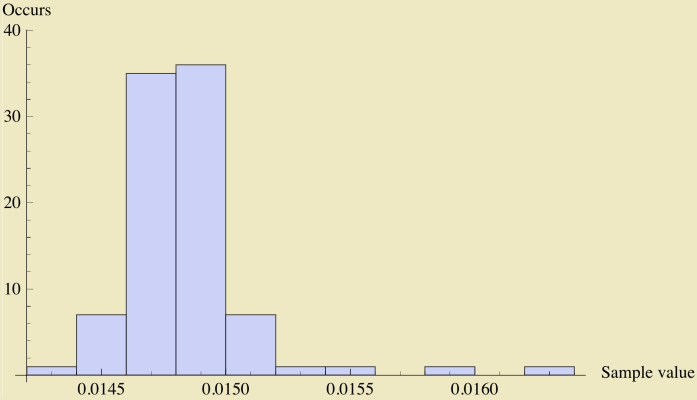


Sample set 3 with 90 values. P-values: St= $4.0158 \times 10^{-6}$  Rt=0.

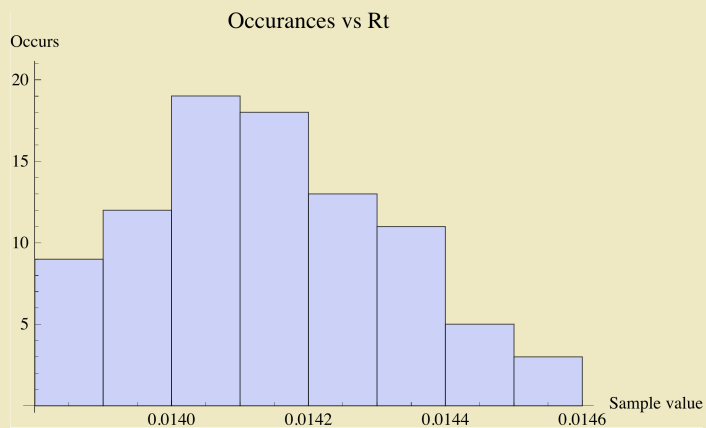
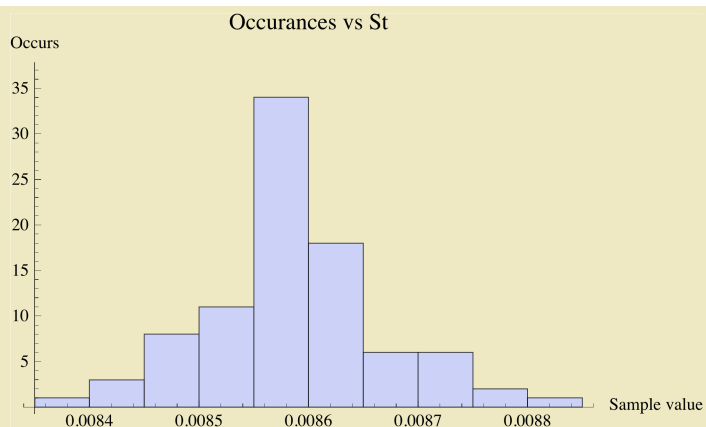
Occurances vs St



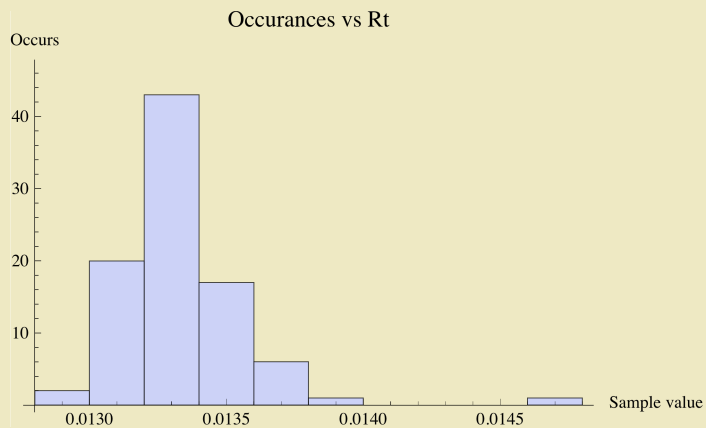
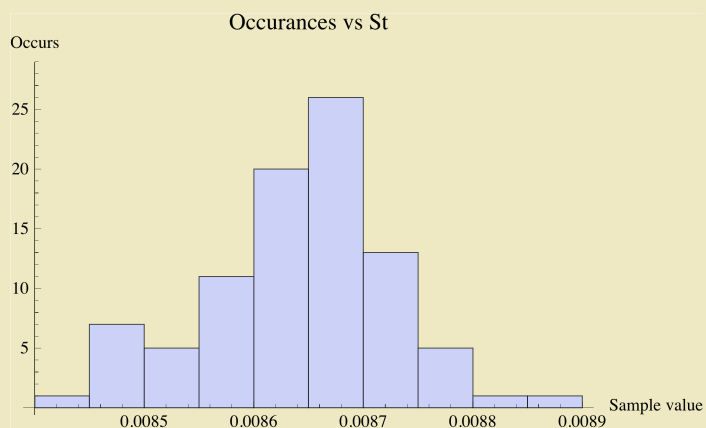
Occurances vs Rt



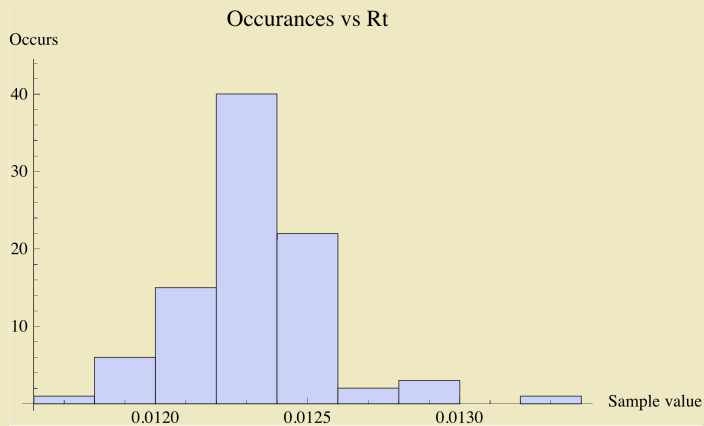
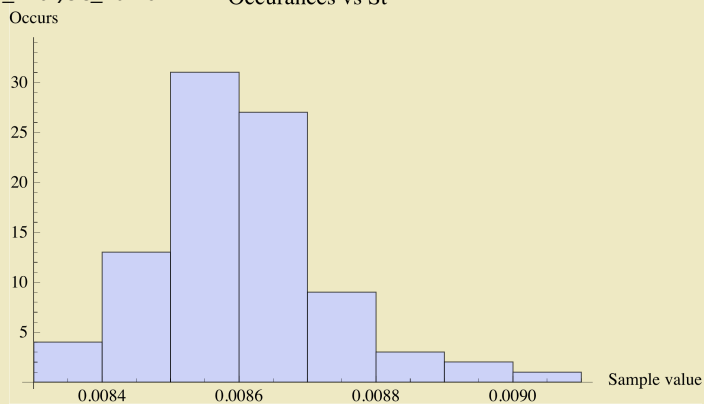
Sample set 4 with 90 values. P-values: St=0.000872209 Rt=0.165478



Sample set 5 with 90 values. P-values: St=0.0575707 Rt=0.000138781



Sample set 6 with 90 values. P-values: St=0.000312928 Rt=0.00206057



-----

This number should be much greater than 0.05: 0.573876 If not try again by re-evaluating.

## Sample Comparison Tests (when normality exists)

Assuming our samples **are normally distributed**, now it's time to see if they are significantly different. If so, then we know changing the number of latches and chains indeed makes a significant performance difference...at least statistically.

The null hypothesis is; there is no real difference between our samples sets. We need to statistically prove that any difference is the result of randomness; like we just happened to pick poor set of samples and it makes their difference look much worse than it really is.

A t-test will produce a statistic p. The p value is a probability, with a value ranging from zero to one. It is the answer to this question: If the populations really have the same mean overall, what is the probability that random sampling would lead to a difference between sample means larger than observed?

For example, if the p value is 0.03 we can say a random sampling from identical populations would lead to a difference smaller than you observed in 97% of the experiments and larger than you observed in 3% of the experiments.

Said another way, suppose I have a single sample set and I copy it, resulting in two identical sample sets. Now suppose we perform a significance test on these two identical sample sets. The resulting p-value will be 1.0 because they are exactly the same. We are essentially doing the same thing here except we have two different sample sets... but we still want to see if they "like" each other..and in our case we hope they are NOT the like each other, which means the p-value will low... below our cut off value of 0.05.

For our analysis we choose alpha of 0.05. To accept that our two samples are statistically similar the p value would need to be less than 0.05 (our alpha).

Good reference about the P-Value and significance testing: <http://www.graphpad.com/articles/pvalue.htm>

Here we go (assuming our samples are normally distributed):

1. Our P value threshold is 0.05, which is our alpha.
2. The null hypothesis is the two populations have the same mean. (Remember we have to sample sets, which not the population.)
3. Do the statistical test to compute the P value.
4. Compare the result P value to our threshold alpha value. If the P value is less then our threshold, we will reject the null hypothesis and say the difference between our samples is significant. However, if the P value is greater than the threshold, we cannot reject the null hypothesis and any difference between our samples are not statistically significant.

In[35]:=

```
Do[
  pValueSt = TTest[{ssSt[i], ssSt[i + 1]]};
  Print["St: (", Length[ssSt[i]],
    " values) pvalue between sample set ", i, " and ", i + 1, " is ", pValueSt];
  pValueRt = TTest[{ssRt[i], ssRt[i + 1]]};
  Print["Rt: (", Length[ssRt[i]],
    " values) pvalue between sample set ", i, " and ", i + 1, " is ", pValueRt];
  ,
  {i, 1, ssNum - 1}
];
```

TTest::nortst : At least one of the p-values in {0.0726837, 0.00024229}, resulting from a test for normality, is below 0.025`. The tests in {T} require that the data is normally distributed. >>

St: (90 values) pvalue between sample set 1 and 2 is  $2.31067 \times 10^{-78}$

TTest::nortst : At least one of the p-values in {0.455729, 0.}, resulting from a test for normality, is below 0.025`. The tests in {T} require that the data is normally distributed. >>

Rt: (90 values) pvalue between sample set 1 and 2 is  $1.97775 \times 10^{-54}$

TTest::nortst : At least one of the p-values in {0.00024229,  $4.0158 \times 10^{-6}$ }, resulting from a test for normality, is below 0.025`. The tests in {T} require that the data is normally distributed. >>

General::stop : Further output of TTest::nortst will be suppressed during this calculation. >>

St: (90 values) pvalue between sample set 2 and 3 is 0.624216

Rt: (90 values) pvalue between sample set 2 and 3 is  $1.98916 \times 10^{-9}$

St: (90 values) pvalue between sample set 3 and 4 is 0.0886415

Rt: (90 values) pvalue between sample set 3 and 4 is  $1.05559 \times 10^{-51}$

St: (90 values) pvalue between sample set 4 and 5 is 0.0000394894

Rt: (90 values) pvalue between sample set 4 and 5 is  $2.1358 \times 10^{-61}$

St: (90 values) pvalue between sample set 5 and 6 is 0.0250373

Rt: (90 values) pvalue between sample set 5 and 6 is  $1.79677 \times 10^{-69}$

If the above T-Test results (p value) are less then our threshold we can say there is a significant difference between the two sample sets.

## Sample Comparison Tests (when normality does NOT exist)

If our sample sets are **not normally distributed**, we can not perform a simple t-test. We can perform what are called location tests. I did some research on significance testing when non-normal distributions exists. I found a very nice reference:

<http://www.statsoft.com/textbook/nonparametric-statistics>

The paragraph below (which is from the reference above) is a key reference to what we're doing here:

...the need is evident for statistical procedures that enable us to process data of "low quality," from small samples, on variables about which nothing is known (concerning their distribution). Specifically, nonparametric methods were developed to be used in cases when the researcher knows nothing about the parameters of the variable of interest in the population (hence the name nonparametric). In more technical terms, nonparametric methods do not rely on the estimation of parameters (such as the mean or the standard deviation) describing the distribution of the variable of interest in the population

Being that I'm not a statistician but still need to determine if these sample sets are significant different, I let *Mathematica* determine the appropriate test. Notice that one of the above mentioned tests will probably be the test *Mathematica* chooses.

Note: If we run our normally distributed data through this analysis (speically, the "LocationEquivalenceTest"), *Mathematica* should detect this and use a more appropriate significant test, like a t-test.

Here we go with the hypothesis testing (assuming our sample sets are not normally distributed):

1. Our P value threshold is 0.05, which is our alpha.
2. The null hypotheses is the two populations have the same mean. (Remember we have to sample sets, which is not the population.)
3. Do the statistical test to compute the P value.
4. Compare the result P value to our threshold alpha value. If the P value is less then our threshold, we will reject the null hypothesis and say the difference between our samples is significant. (Which is what I'm hoping to see.) However, if the P value is greater than the threshold, we cannot reject the null hypothesis and any difference between our samples are not statistically significant; randomness, picked the "wrong" samples, etc.

In[36]:=

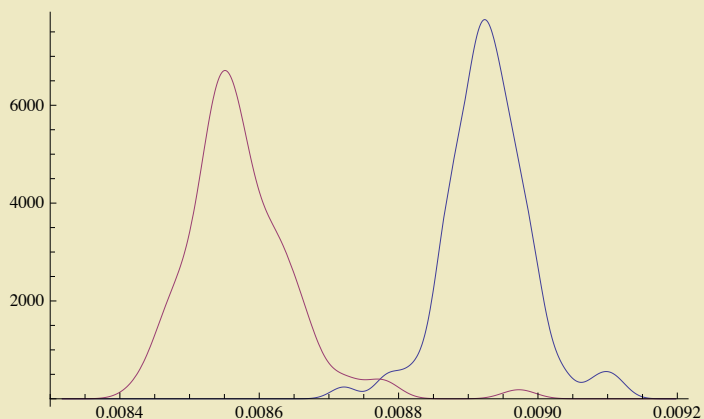
```
Do[
  StHist = SmoothHistogram[{ssSt[i], ssSt[i + 1]}];
  StTest1 = MannWhitneyTest[{ssSt[i], ssSt[i + 1]}];
  StTest2 = LocationEquivalenceTest[{ssSt[i], ssSt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["St: (", Length[ssSt[i]], " values) Between sample ",
    i, " and ", i + 1, ". Test1=", StTest1, " Test2=", StTest2];
  Print[StHist];
  Print["-----"];
  QtHist = SmoothHistogram[{ssQt[i], ssQt[i + 1]}];
  QtTest1 = MannWhitneyTest[{ssQt[i], ssQt[i + 1]}];
  QtTest2 = LocationEquivalenceTest[{ssQt[i], ssQt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["Qt: (", Length[ssQt[i]], " values) Between sample ",
    i, " and ", i + 1, ". Test1=", QtTest1, " Test2=", QtTest2];
  Print[QtHist];
  Print["-----"];
  RtHist = SmoothHistogram[{ssRt[i], ssRt[i + 1]}];
  RtTest1 = MannWhitneyTest[{ssRt[i], ssRt[i + 1]}];
  RtTest2 = LocationEquivalenceTest[{ssRt[i], ssRt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["Rt: (", Length[ssRt[i]], " values) Between sample ",
    i, " and ", i + 1, ". Test1=", RtTest1, " Test2=", RtTest2];
  Print[RtHist];
  Print[
    "-----"
    ---"];
, {i, 1, ssNum - 1}
];
```

St: (90 values) Between sample 1 and 2. Test1=

$6.03913 \times 10^{-30}$  Test2={

	Statistic	P-Value
Kruskal-Wallis	129.263	$2.2215 \times 10^{-51}$

, KruskalWallis}



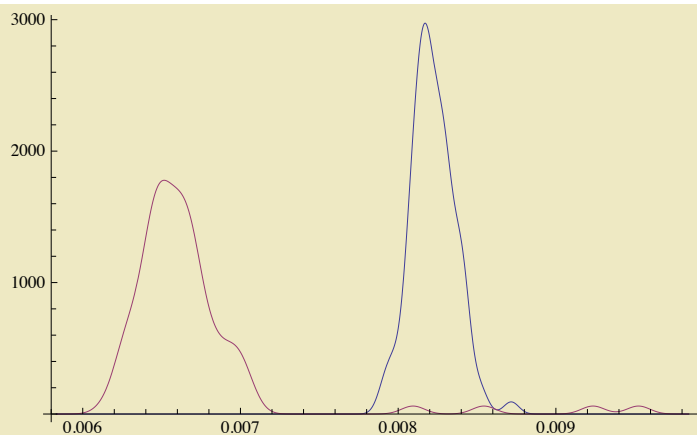
Qt: (90 values) Between sample 1 and 2. Test1=

$4.48201 \times 10^{-27}$  Test2={

	Statistic	P-Value
Kruskal-Wallis	116.147	$2.60432 \times 10^{-42}$

, KruskalWallis}



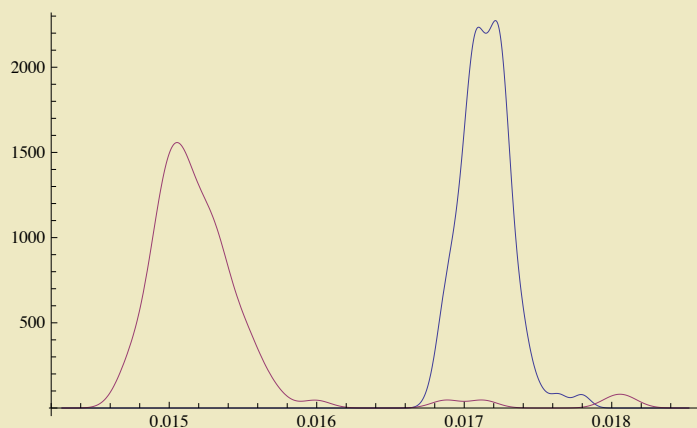


Rt: (90 values) Between sample 1 and 2. Test1=

$7.75932 \times 10^{-28}$  Test2= { 

	Statistic	P-Value
Kruskal-Wallis	119.626	$1.61613 \times 10^{-44}$

 , KruskalWallis }

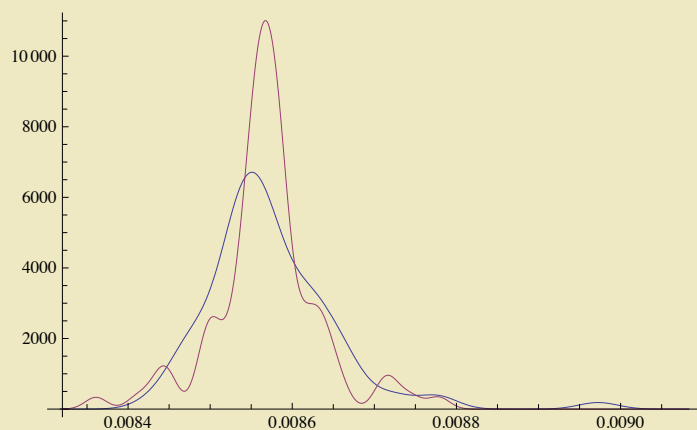


St: (90 values) Between sample 2 and 3. Test1=

0.658478 Test2= { 

	Statistic	P-Value
Kruskal-Wallis	0.194115	0.660767

 , KruskalWallis }

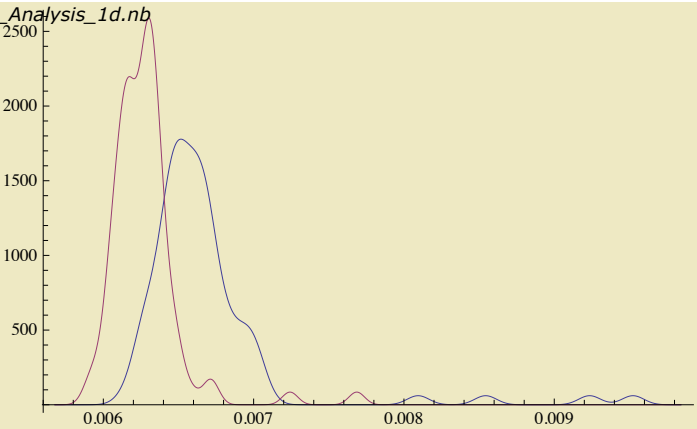


Qt: (90 values) Between sample 2 and 3. Test1=

$4.57456 \times 10^{-20}$  Test2= { 

	Statistic	P-Value
Kruskal-Wallis	84.1813	$2.38184 \times 10^{-26}$

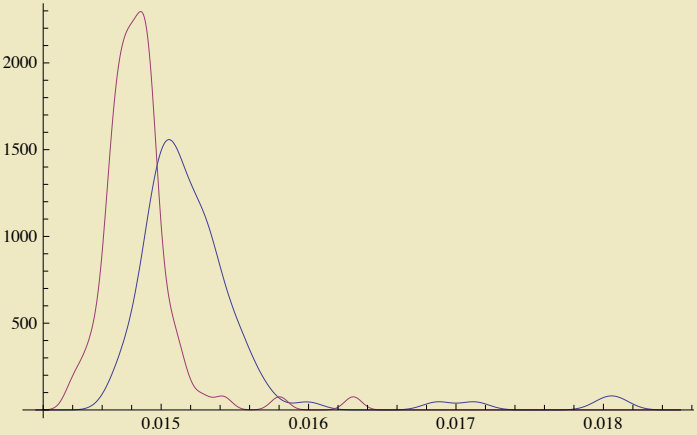
 , KruskalWallis }



Rt: (90 values) Between sample 2 and 3. Test1=  
 $4.77441 \times 10^{-18}$  Test2={

	Statistic	P-Value
Kruskal-Wallis	74.9968	$9.44045 \times 10^{-23}$

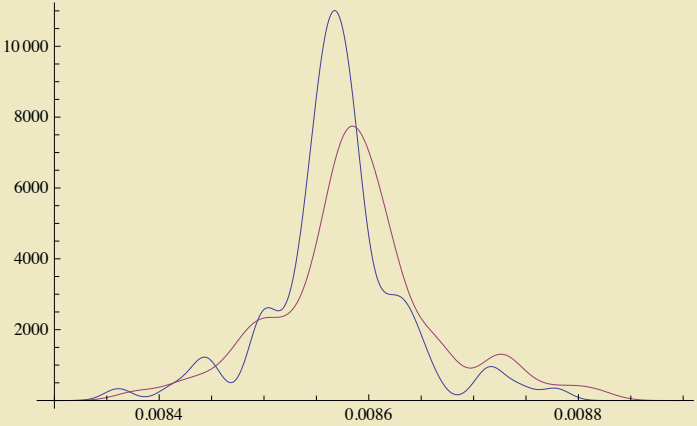
, KruskalWallis}



St: (90 values) Between sample 3 and 4. Test1=  
0.0304412 Test2={

	Statistic	P-Value
Kruskal-Wallis	4.67801	0.030153

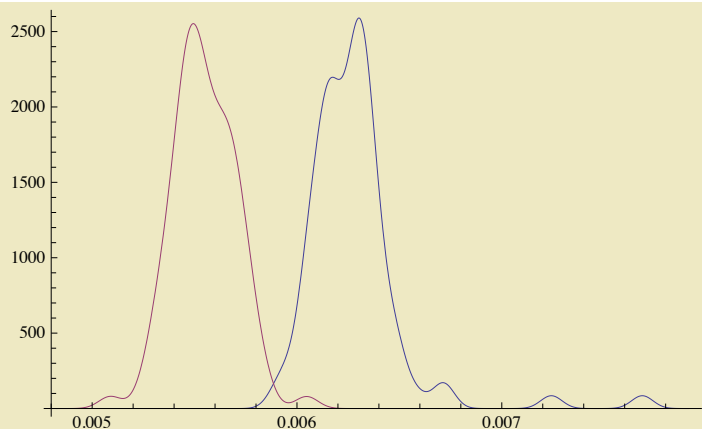
, KruskalWallis}



Qt: (90 values) Between sample 3 and 4. Test1=  
 $5.97135 \times 10^{-31}$  Test2={

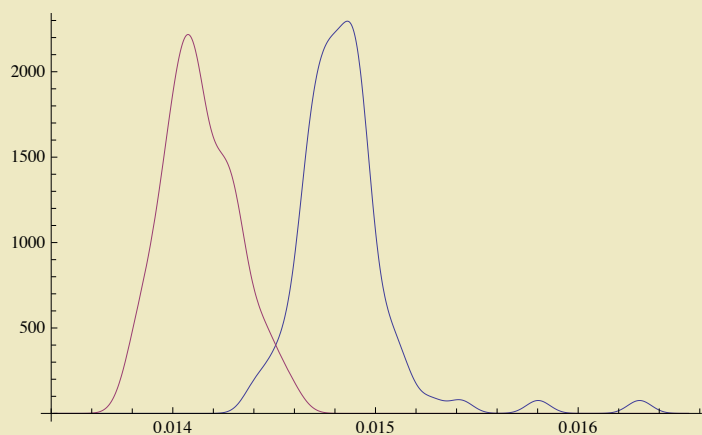
	Statistic	P-Value
Kruskal-Wallis	133.857	$3.92053 \times 10^{-55}$

, KruskalWallis}



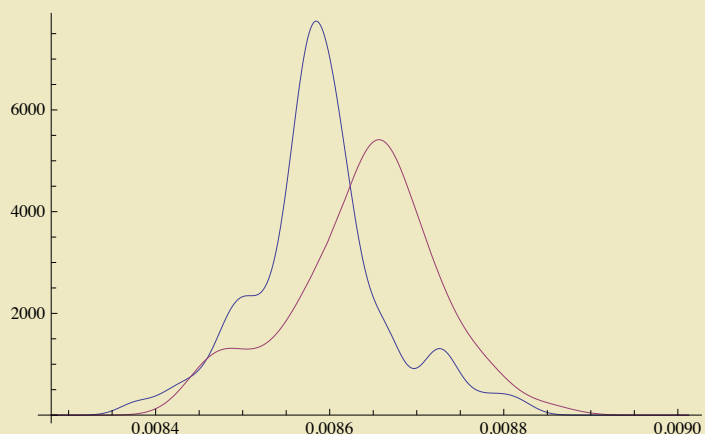
Rt: (90 values) Between sample 3 and 4. Test1=

$$1.20049 \times 10^{-30} \quad \text{Test2} = \left\{ \begin{array}{cc|cc} & & \text{Statistic} & \text{P-Value} \\ \hline \text{Kruskal-Wallis} & & 132.47 & 5.81996 \times 10^{-54} \end{array} \right\}, \text{KruskalWallis}$$



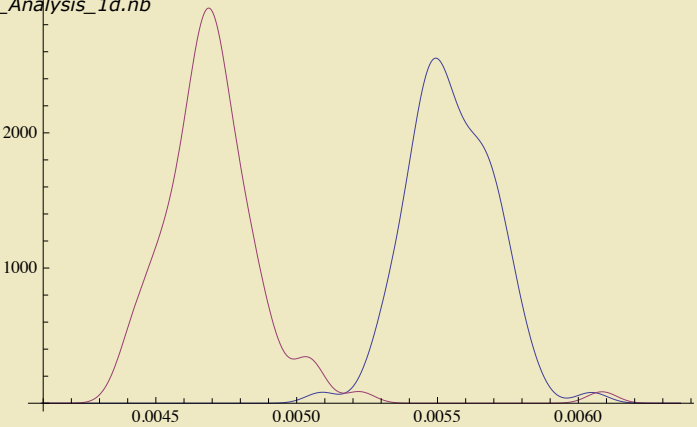
St: (90 values) Between sample 4 and 5. Test1=

$$4.60999 \times 10^{-6} \quad \text{Test2} = \left\{ \begin{array}{cc|cc} & & \text{Statistic} & \text{P-Value} \\ \hline \text{Kruskal-Wallis} & & 20.9798 & 2.55016 \times 10^{-6} \end{array} \right\}, \text{KruskalWallis}$$



Qt: (90 values) Between sample 4 and 5. Test1=

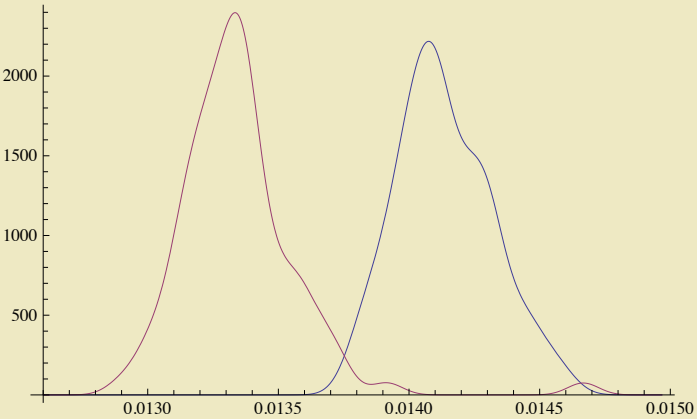
$$9.86417 \times 10^{-30} \quad \text{Test2} = \left\{ \begin{array}{cc|cc} & & \text{Statistic} & \text{P-Value} \\ \hline \text{Kruskal-Wallis} & & 128.289 & 1.25271 \times 10^{-50} \end{array} \right\}, \text{KruskalWallis}$$



Rt: (90 values) Between sample 4 and 5. Test1=  
1.28051 × 10<sup>-29</sup> Test2={

	Statistic	P-Value
Kruskal-Wallis	127.771	3.10093 × 10 <sup>-50</sup>

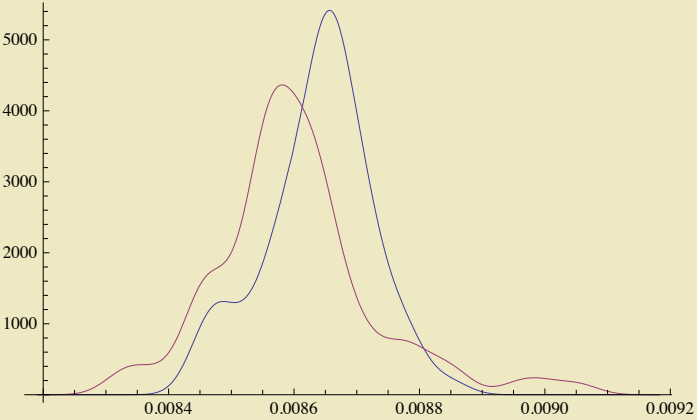
, KruskalWallis}



St: (90 values) Between sample 5 and 6. Test1=  
0.000587284 Test2={

	Statistic	P-Value
Kruskal-Wallis	11.8257	0.000495599

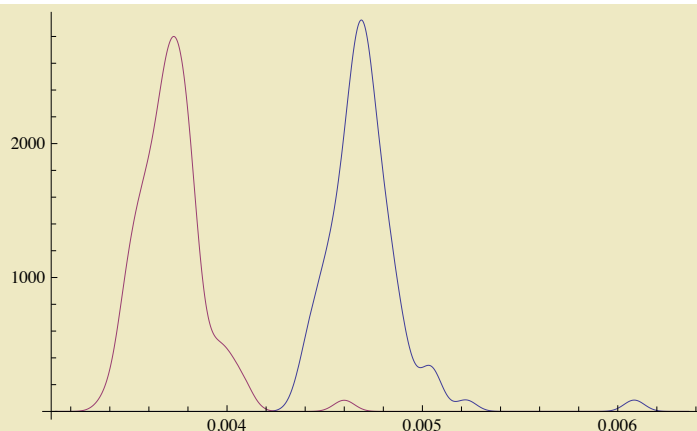
, KruskalWallis}



Qt: (90 values) Between sample 5 and 6. Test1=  
1.05126 × 10<sup>-30</sup> Test2={

	Statistic	P-Value
Kruskal-Wallis	132.734	3.50702 × 10 <sup>-54</sup>

, KruskalWallis}

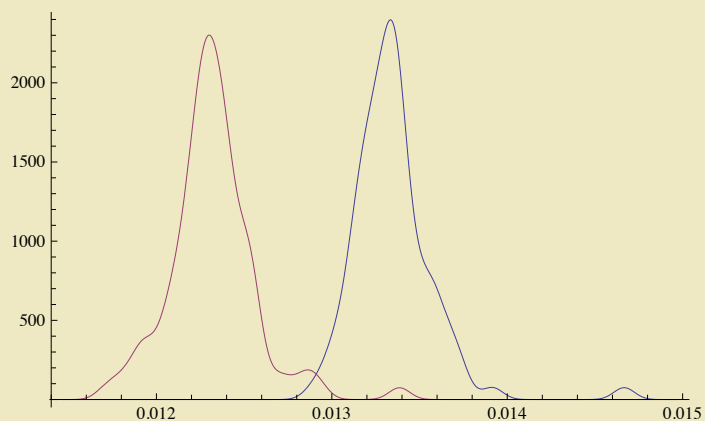


Rt: (90 values) Between sample 5 and 6. Test1=

$3.94146 \times 10^{-30}$  Test2= { 

Statistic	P-Value
Kruskal-Wallis	130.11

 $4.80047 \times 10^{-52}$  , KruskalWallis }



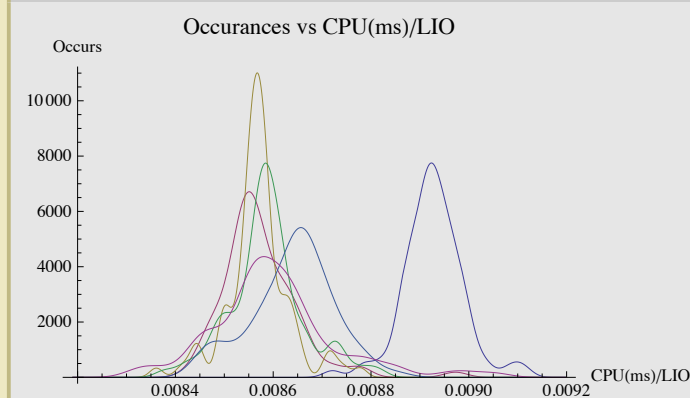
## Visually Comparing All Samples

I also wanted to get a nice visual picture of my sample sets...together. Sometimes I include all the sample sets and sometimes I don't. It's just based on what I want to convey. Sometimes you get a more appropriate view if all the data is not included. That is, remove ssSt[6] for example.

In[37]:=

```
SmoothHistogram[{ssSt[1], ssSt[2], ssSt[3], ssSt[4], ssSt[5], ssSt[6]},
  PlotLabel -> "Occurrences vs CPU(ms)/LIO", AxesLabel -> {"CPU(ms)/LIO", "Occurs"}]
```

Out[37]=

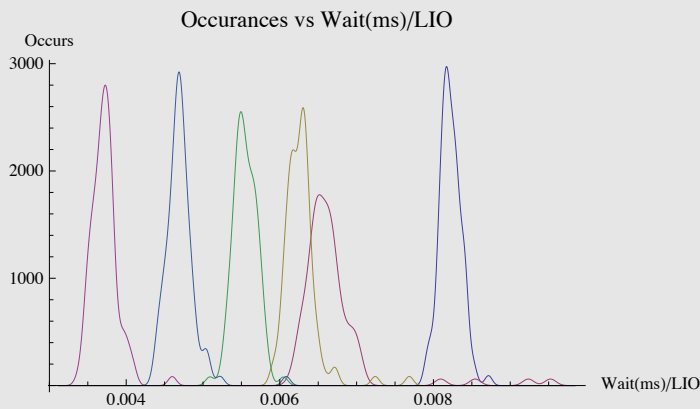




In[38]:=

```
SmoothHistogram[{ssQt[1], ssQt[2], ssQt[3], ssQt[4], ssQt[5], ssQt[6]},
  PlotLabel -> "Occurances vs Wait(ms)/LIO", AxesLabel -> {"Wait (ms) /LIO", "Occurs"}]
```

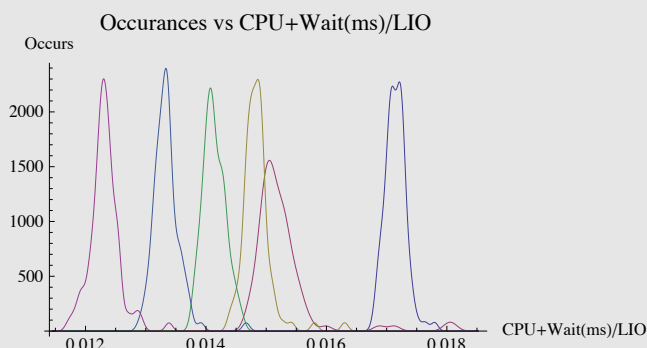
Out[38]=



In[39]:=

```
SmoothHistogram[{ssRt[1], ssRt[2], ssRt[3], ssRt[4], ssRt[5], ssRt[6]},
  PlotLabel -> "Occurances vs CPU+Wait(ms)/LIO", AxesLabel -> {"CPU+Wait (ms) /LIO", "Occurs"}]
```

Out[39]=



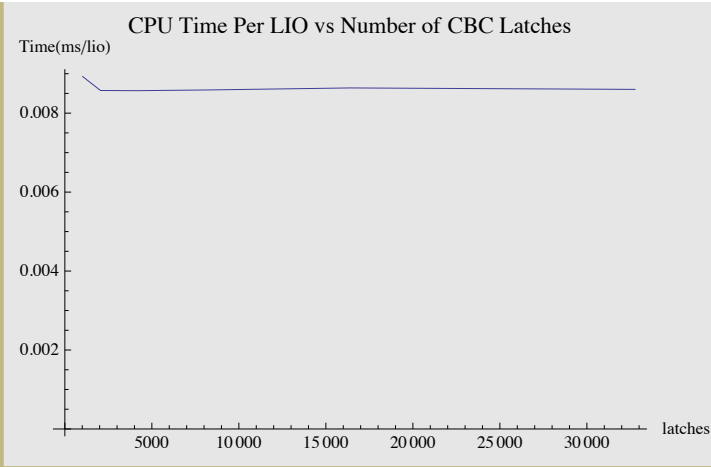
## Important Graphics

I also wanted to get a nice visual picture of my sample sets...together. Sometimes I include all the sample sets and sometimes I don't... setting the leaveOut variable easily controls this. It's just based on what I want to convey.

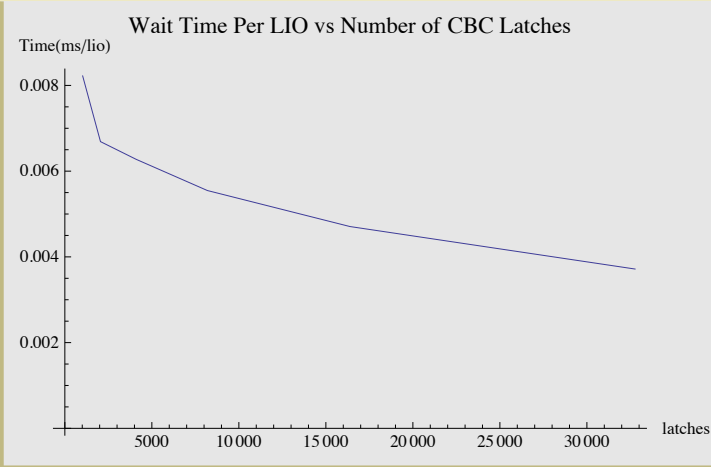
In[40]:=

```
leaveOut = 0;
St = Table[
  {latches[i], Mean[ssSt[i]]}
  , {i, ssNum - leaveOut}
];
Qt = Table[
  {latches[i], Mean[ssQt[i]]}
  , {i, ssNum - leaveOut}
];
Rt = Table[
  {latches[i], Mean[ssSt[i]] + Mean[ssQt[i]]}
  , {i, ssNum - leaveOut}
];
ListPlot[St, AxesOrigin -> {0, 0}, PlotStyle -> PointSize[Large], Joined -> True,
  AxesLabel -> {"latches", "Time(ms/lio)"}, PlotLabel -> "CPU Time Per LIO vs Number of CBC Latches"]
ListPlot[Qt, AxesOrigin -> {0, 0}, PlotStyle -> PointSize[Large], Joined -> True,
  AxesLabel -> {"latches", "Time(ms/lio)"}, PlotLabel -> "Wait Time Per LIO vs Number of CBC Latches"]
ListPlot[{St, Rt}, AxesOrigin -> {0, 0}, PlotStyle -> PointSize[Large],
  Joined -> True, AxesLabel -> {"latches", "Time(ms/lio)"},
  PlotLabel -> "CPU+Wait Time Per LIO vs Number of CBC Latches"]
```

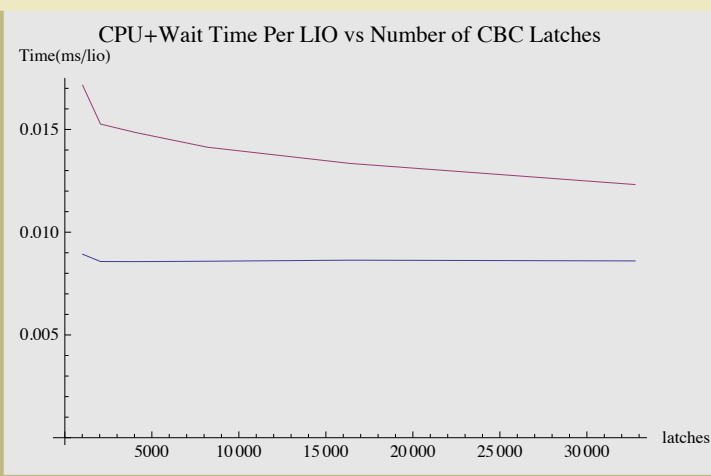
Out[44]=



Out[45]=



Out[46]=



In[47]=

```
leaveOut = 2;  
ListPlot[  
  Table[  
    {{latches[i], Mean[ssSt[i]]},  
     {latches[i], Mean[ssSt[i]] + Mean[ssQt[i]}}  
    ], {i, ssNum - leaveOut}  
], AxesOrigin -> {0, 0}, PlotStyle -> PointSize[Large],  
Joined -> True, AxesLabel -> {"latches", "Time(ms)"}  
]
```

Out[48]=

