SAND2017-2862C

13th ANNUAL WORKSHOP 2017

LDMS AND INFINIBAND @ SANDIA

Serge Polevitzky, SAIC HPC Support at SNL

SAIC

[March, 2017]

INTRODUCTION

- **What You Should Expect**
  - Some Disclosures: System Described, Presupposition, *etc*.
  - Next, Why We Started Our Investigation
  - Background on Tools Used
  - What We've Seen (The Recent Story)
  - Expect "Support" from the Audience ! [Interrupt to Correct, to Add Info, *etc*. … in short, PLEASE PARTICIPATE]
  - We Will Follow the old Cray Research Mantra of "We Don't Take Ourselves Seriously, But We Take What We Do Very Seriously."

THE PROBLEM / THE "INSURMOUNTABLE OPPORTUNITY"

FABRIC MELTDOWN

- **One (More ?) IB-Connected Clusters Had Meltdowns**
  - Started w/ Experiencing an Occasional IB “Meltdown”
    [“Cure” Is/Was Pretty Draconian]
  - We Knew of Livermore’s “Sniper Script” Solution, But ...
  - Didn’t Have Sufficient Evidence as to Whom to “Snipe”
  - Tribal Knowledge Was That It Was Our LNET Routers
  - Frequency of Meltdowns Started to Increase ....
  - A “Must Resolve This ***Now***” Edict Was Issued

FABRIC MELTDOWN (CONT'D)

- **Relocated & *Reduced* Number of LNET Gateways (…?) &**
- **Relocated the Subnet Manager (SM)**
- **An *a priori* Solution ….**
- **And It Worked ! Life Got a Lot Better. We Took Stock.**
- **Our "Tool Kit": Smart People (*coup d'oeil*) !, perfquery, PerfMgr, LDMS**
  - perfquery (But Many Functions are ***optional*** !?)
  - PerfMgr, But the Data Wasn't "Helpful"
  - LDMS. Just Starting to Be Used, so Little "Street Cred"

FABRIC MELTDOWN (CONT'D)

- **~ 18 perfquery commands are "optional" (YMMV)**
  - -D, --xmtdisc show transmit discard details. This is an optional counter.
  - -E, --rcverr show receive error details. This is an optional counter.
  - --slrcvfecn show SL Rcv FECN counters. This is an optional counter.
  - --slrcvbecn show SL Rcv BECN counters. This is an optional counter.
  - perfquery –rcvcc PortRcvConCtrl perfquery: iberror: failed: cannot query

FABRIC MELTDOWN (CONT'D)

- **We "Benched" perfquery. This Left PerfMgr & LDMS.**
- **Three "Sticking Points" Emerged w/ PerfMgr & LDMS.**
  1) PerfMgr found 1 Link_Downed; LDMS Saw Many,
  2) PerfMgr Saw Many VL15 Drops; LDMS Saw 0,
  3) PerfMgr Saw 0 port_transmit_waits; LDMS Saw Many.

- **port_transmit_wait :: *Great Indicator of Congestion* ! The Canary in the Coal Mine (for congestion).**
- **Let's Start w/** 1) LDMS Sees Many Link_Downed Counts

FABRIC MELTDOWN (CONT'D)

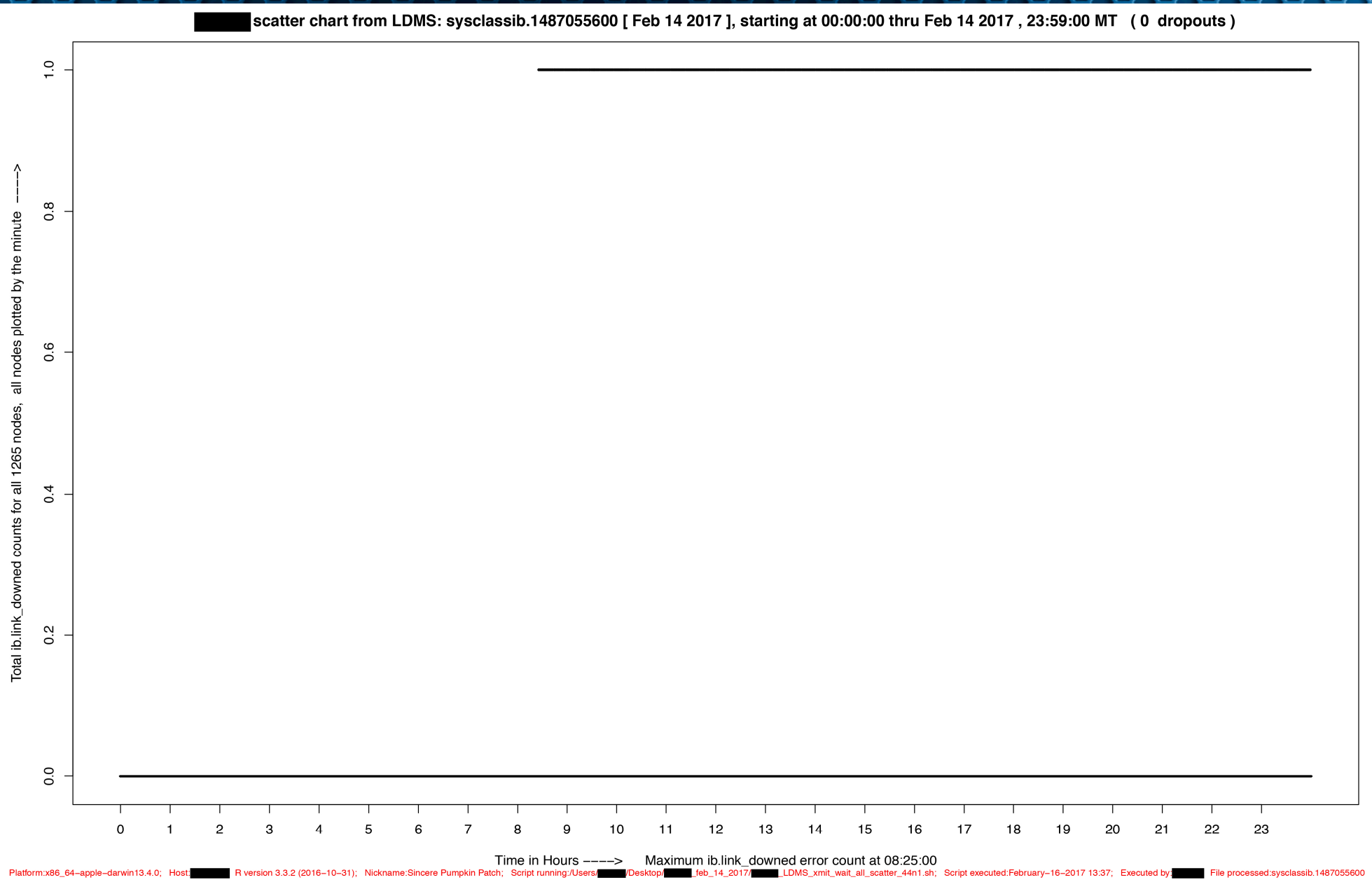
- **Link_Downed Counts: PerfMgr & LDMS …**
  - Both Saw the Link_Downed Happening at the Same Time (±)
  - LDMS Sees the Link_Downed @ 08:25 (505 minutes after 12:00)
  - Why is 505 Minutes After Midnight Important ? [See Next Chart]

    > count_ib.link_downed <- sum(data_set$ib.link_downed)

    > count_ib.link_downed

    > [1] 935

FABRIC MELTDOWN (CONT'D)



FABRIC MELTDOWN (CONT'D)

- **Invoking the Willing Suspension of Disbelief …**
  - PerfMgr Saw 1 Link_Downed [the Court accepts this Claim]
  - Straight from LDMS .csv file for Link_Downed Event →
    - ✓ Time … CompID … ib.link_downed
    - ✓ 1487085900** 120007 1
  - REDACTED-HOST-NAME-login[1-8] ldmsd_idbase=120000 [login node 7 Had the 1 Link_Downed Count]
  - Submit to the Court that PerfMgr & LDMS Agree; LDMS PerfMgr Just Reporting the Data Differently
- **Sticking Point #1 Resolved …**

** 2/14/2017, 8:25:00 AM

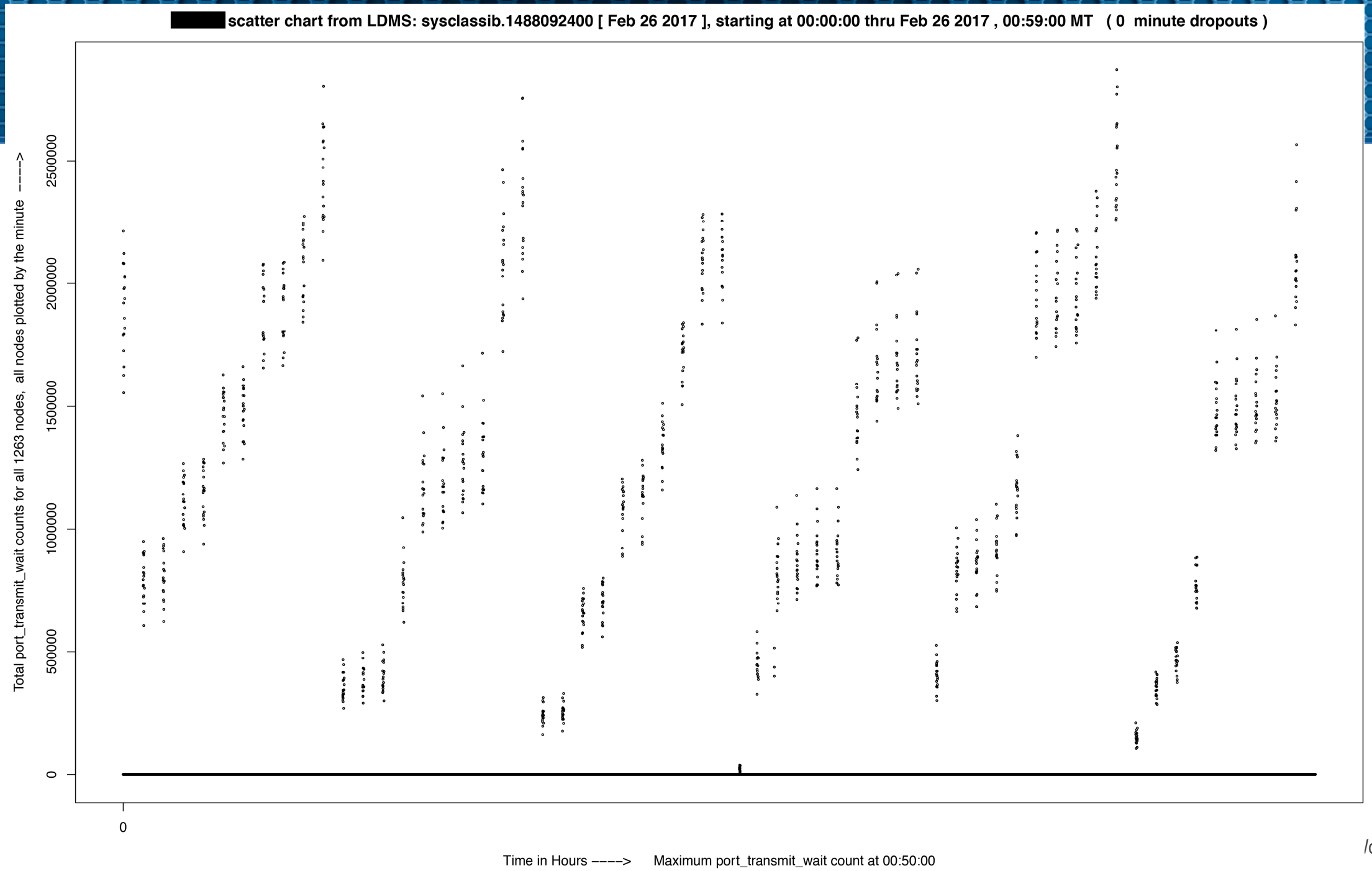
FABRIC MELTDOWN (CONT'D)

- **Continuing the with the Willing Suspension of Disbelief:**
- **Sticking Point #2 (PerfMgr Sees Many VL15s Dropped …) and LDMS Sees ZERO**
- **PerfMgr Looks at *Switches and* HCAs ! LDMS Only Looks at HCAs (at Least at the Time of These Data Captures)**
  1) How Can an HCA Know It has Dropped a VL15 ?
  2) Conclusion: Since PerfMgr Sees Switch Info, "Naturally" PerfMgr Will See the VL15s Dropped & LDMS Will Not
- **Submit that Sticking Point #2 Is Resolved**

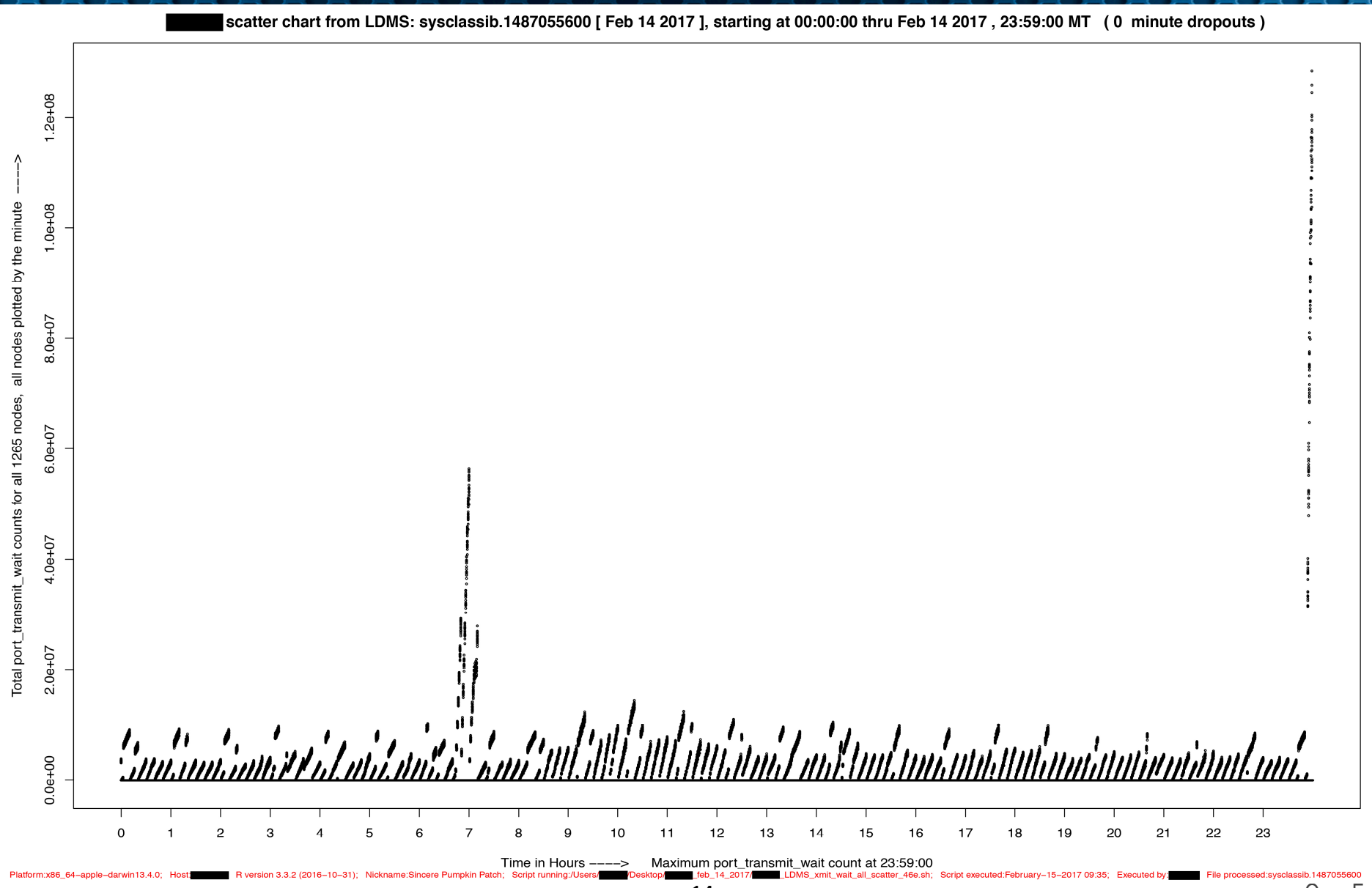
FABRIC MELTDOWN (CONT'D)

- **More Willing Suspension of Disbelief Now Required:**
- **PerfMgr Sees ZERO port_transmit_waits (from Any Source)**
  - LDMS Shows → Next Slide [a One-Hour Capture"]
  - ✓ Whatever It Is, It Gets Reset Every 10-Minutes
  - ✓ Some (Possibly Small) Increment Every Minute
  - ✓ The Counts Are Non-Zero (so Disagreement w/ PerfMgr)

Next Slide → {Many port_transmit_waits} // PerfMgr Sees 0


[redacted] scatter chart from LDMS: sysclassib.1488092400 [Feb 26 2017], starting at 00:00:00 thru Feb 26 2017 , 00:59:00 MT (0 minute dropouts)
Total port_transmit_wait counts for all 1263 nodes, all nodes plotted by the minute ----->
0
500000
1000000
1500000
2000000
2500000
0
Time in Hours ----->
Maximum port_transmit_wait count at 00:50:00

FABRIC MELTDOWN (CONT'D)



FABRIC MELTDOWN (CONT'D)

- **PerfMgr Has a Default On/Off Switch & A Threshold Switch Too**
  1) PerfMgr Has a Primary Switch that says "Don't Report Anything"
  2) "perfmgr_xmit_wait_log", OPT_OFFSET(perfmgr_xmit_wait_log), opts_parse_boolean, NULL, 0 }
  3) perfmgr_xmit_wait_log wasn't defined in our opensm.conf
  4) So PerfMgr (for us) Was Blind to All port_transmit_waits
  5) And a 2º Switch, a Threshold for "Do Not Report Count if Less Than X" [#FFFF]
  6) We Then Turned the Switch to "On," and Set the Threshold to 0
- **And ....**

FABRIC MELTDOWN (CONT'D)

- **PerfMgr (cont'd)**
  7) No Change ! Still Seeing 0 port_transmit_waits from PerfMgr …
  8) Still Seeing large Number of port_transmit_waits from LDMS
  9) SME #1: port_transmit_wait Counts Turned Off Because the Counts Were so Large & It's a 32-bit Counter …
  10) SME #2: "The scatter plot looks reasonable for port_transmit_ wait counts"
  11) Hmmm … ??
  12) For the Moment, Suggest Leaning Toward Assuming the LDMS Data Has Some Merit
- **Sticking Point #3 Still NOT QUITE Resolved.**

FABRIC MELTDOWN (CONT'D)

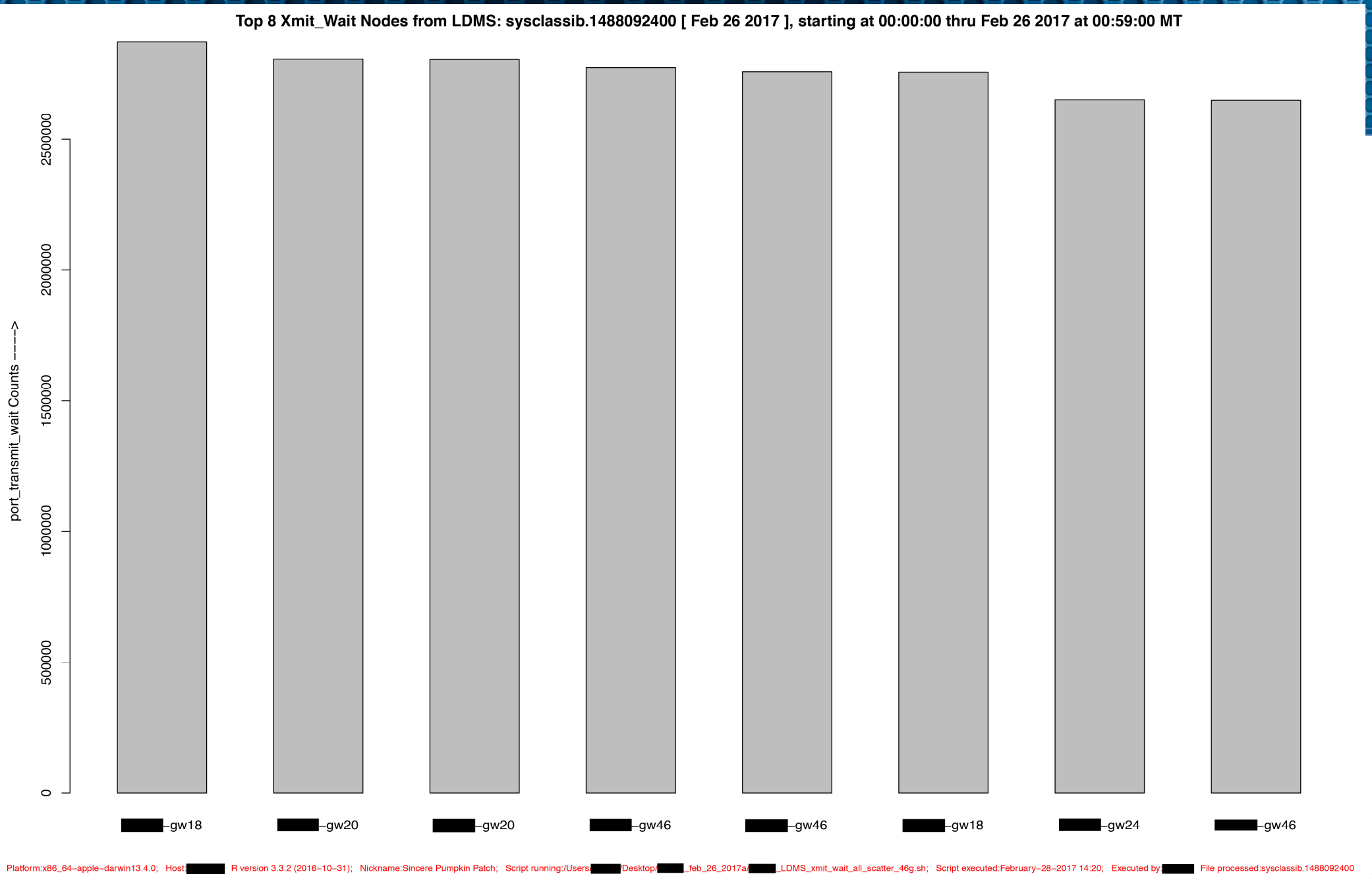
- Still Don't Know What Action to Take for Given Any Given port_transmit_wait Counts, Even If We Were to Have Total Faith in Those Counts from Either PerfMgr and/or LDMS.
- Again, port_transmit_wait Is the Canary in the Coalmine for Congestion, so Monitoring It Would Seem to Be Very Important
- "In God We Trust. All Others Must Bring [Relevant] Data."
- This IB Exploration Is Very Much a Work in Progress. But Here Is a Brief "exploration" of LDMS Data from the One Sandia HPC Cluster

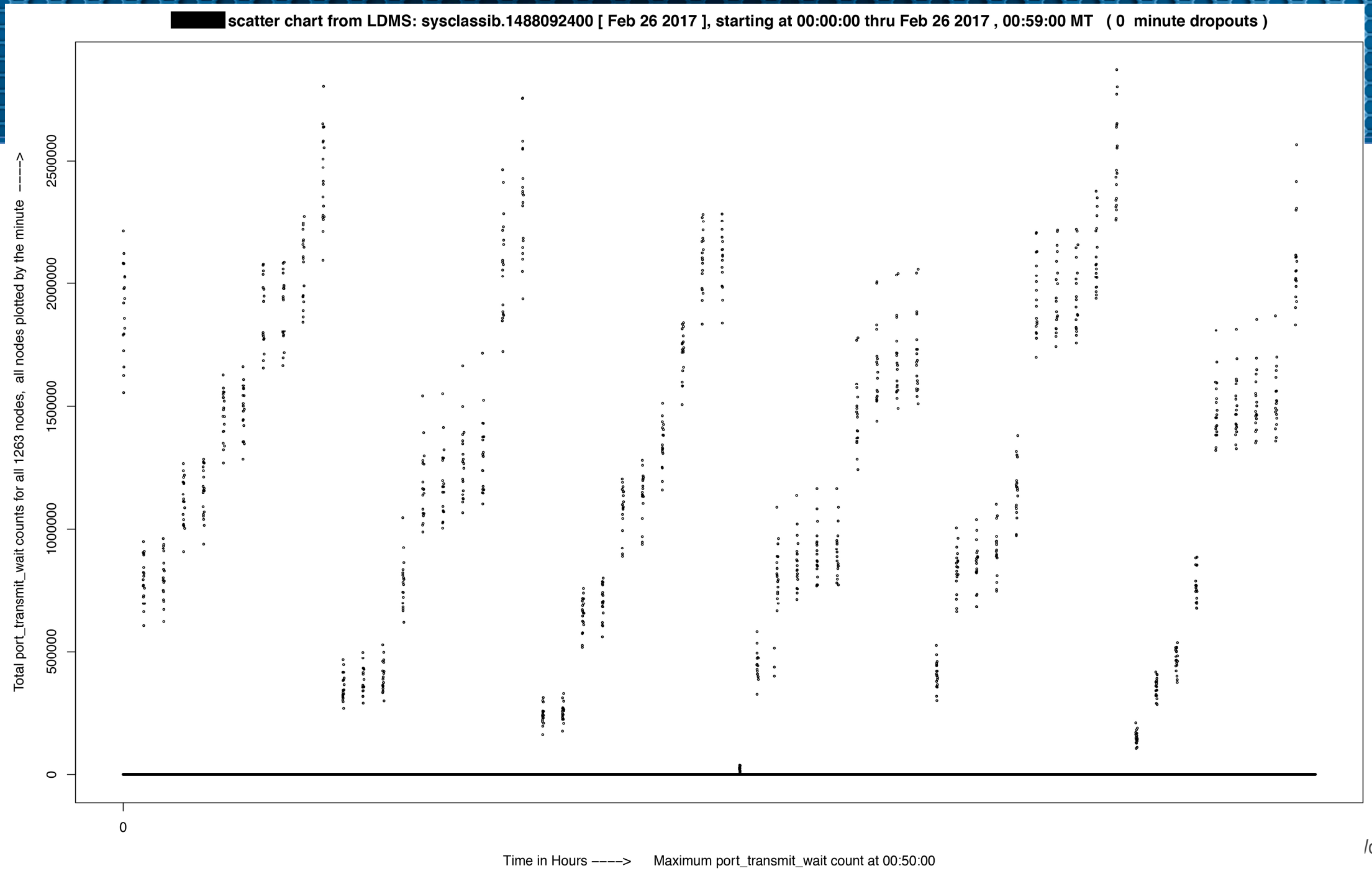
LOOKING AT THE DATA (GIVEN WHAT WE KNOW)

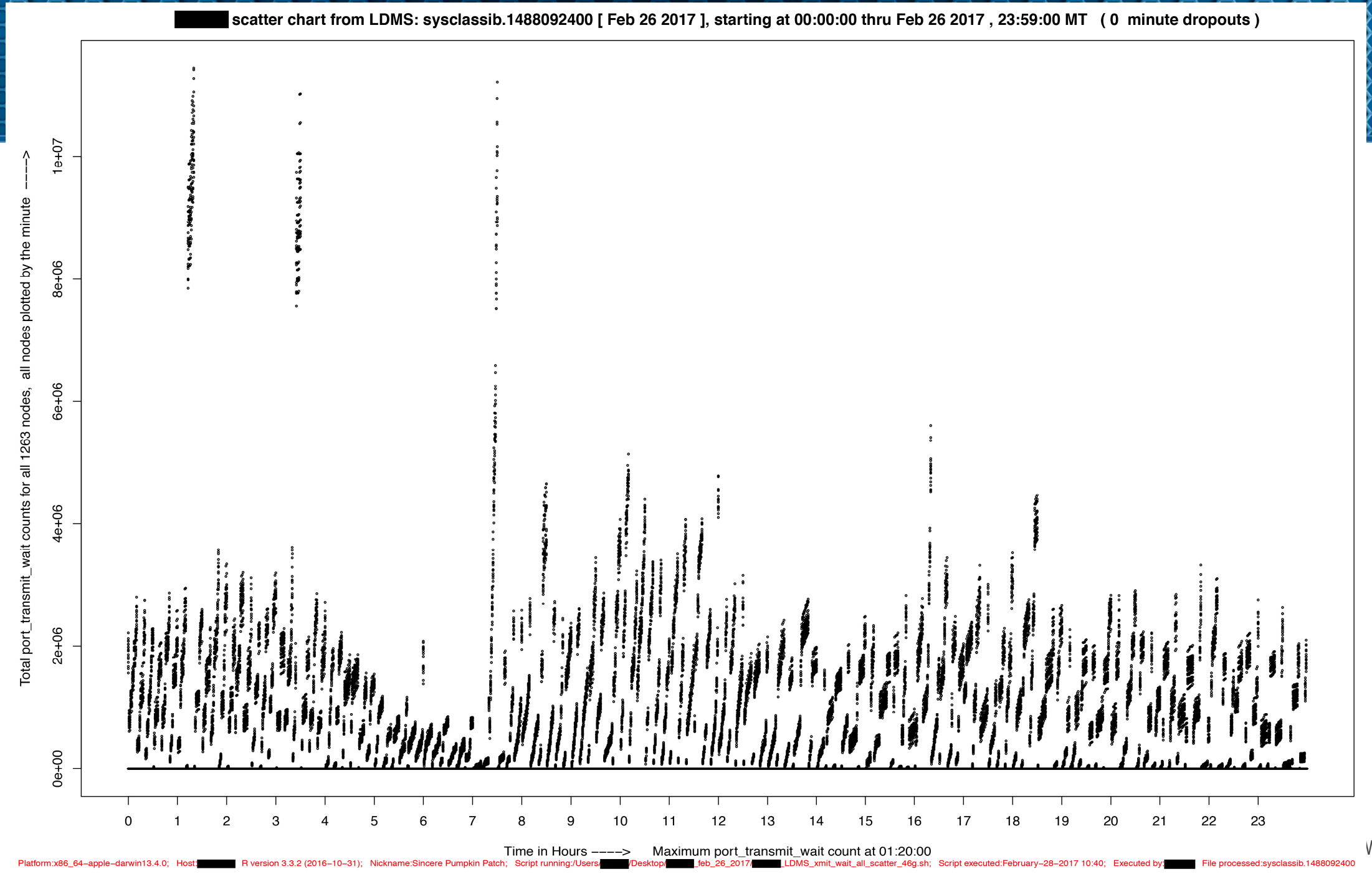
IN AN IMPERFECT WORLD …

- **Suggested Strategy**
  - First, See Any IB Errors ? If so, Attend to Them.
  - If No Errors, Look at Performance, “Got Congestion ?”
  - Performance & Congestion May Take Some Delving into, Not as Clear-Cut a Process as with Error Checking

Platform:x86_64−apple−darwin13.4.0; Host:[redacted] R version 3.3.2 (2016−10−31); Nickname:Sincere Pumpkin Patch; Script running:/Users/[redacted]/Desktop/[redacted]_jan_2017/[redacted]_LDMS_xmit_wait_all_scatter_46e.sh; Script executed:February−15−2017 13:23; Executed by:[redacted] File processed:sysclassib.1479452400


Top 8 Xmit_Wait Nodes from LDMS: sysclassib.1488092400 [Feb 26 2017], starting at 00:00:00 thru Feb 26 2017 at 00:59:00 MT
port_transmit_wait Counts ----->
0
500000
1000000
1500000
2000000
2500000
[redacted]-gw18
[redacted]-gw20
[redacted]-gw20
[redacted]-gw46
[redacted]-gw46
[redacted]-gw18
[redacted]-gw24
[redacted]-gw46
Platform:x86_64-apple-darwin13.4.0; Host:[redacted] R version 3.3.2 (2016-10-31); Nickname:Sincere Pumpkin Patch; Script running:/Users/[redacted]/Desktop/[redacted]_feb_26_2017a/[redacted]_LDMS_xmit_wait_all_scatter_46g.sh; Script executed:February-28-2017 14:20; Executed by:[redacted] File processed:sysclassib.1488092400


[redacted] scatter chart from LDMS: sysclassib.1488092400 [Feb 26 2017], starting at 00:00:00 thru Feb 26 2017 , 00:59:00 MT (0 minute dropouts)
Total port_transmit_wait counts for all 1263 nodes, all nodes plotted by the minute ----->
0
500000
1000000
1500000
2000000
2500000
0
Time in Hours ----->
Maximum port_transmit_wait count at 00:50:00



Vorkshop 2017

CONCLUSION - 1

- **Our Understanding of IB Is Still "New" …**
  - Spread SM, LDMS, & LNET GWs Across Switches, Nodes
  - If Gateway Congestion, Increase Receive Buffers (4X ?)
  - Increase the Number of VLs … [Decreases Each VL's Buffer]
  - More LNET Gateways Probably Better than Fewer
  - Use Both PerfMgr & LDMS
  - Use SPLUNK [Our PerfMgr & LDMS Feed into SPLUNK]

CONCLUSION - 2

- **Some "Possibles" & "Requests"**
  - Have a Metric with Known Performance Characteristics (port_transmit_wait Numbers); On Bring Up, Run a Quick Job that Generates PerfMgr & LDMS Current Values
  - Request a Rosetta Stone [*e.g.*, Clearly Define All Variables [What Is/Are "ib.COUNTER_SELECT2_F", "port_xmit_constraint_errors," *et al*. ?]
  - Request Definitions of "Hidden Info" — *e.g.*, "Alert ! the port_transmit_wait Switch Is Set to NOT Report Data"; "Alert! port_transmit_wait Values > X Usually Result in a port_transmit_discard," *etc*.

OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

THANK YOU ! … ANY QUESTIONS / DISCUSSION ?

Serge Polevitzky

SAIC @ SNL

13th ANNUAL WORKSHOP 2017

BACKUP INFORMATION

Serge Polevitzky

SAIC @ SNL

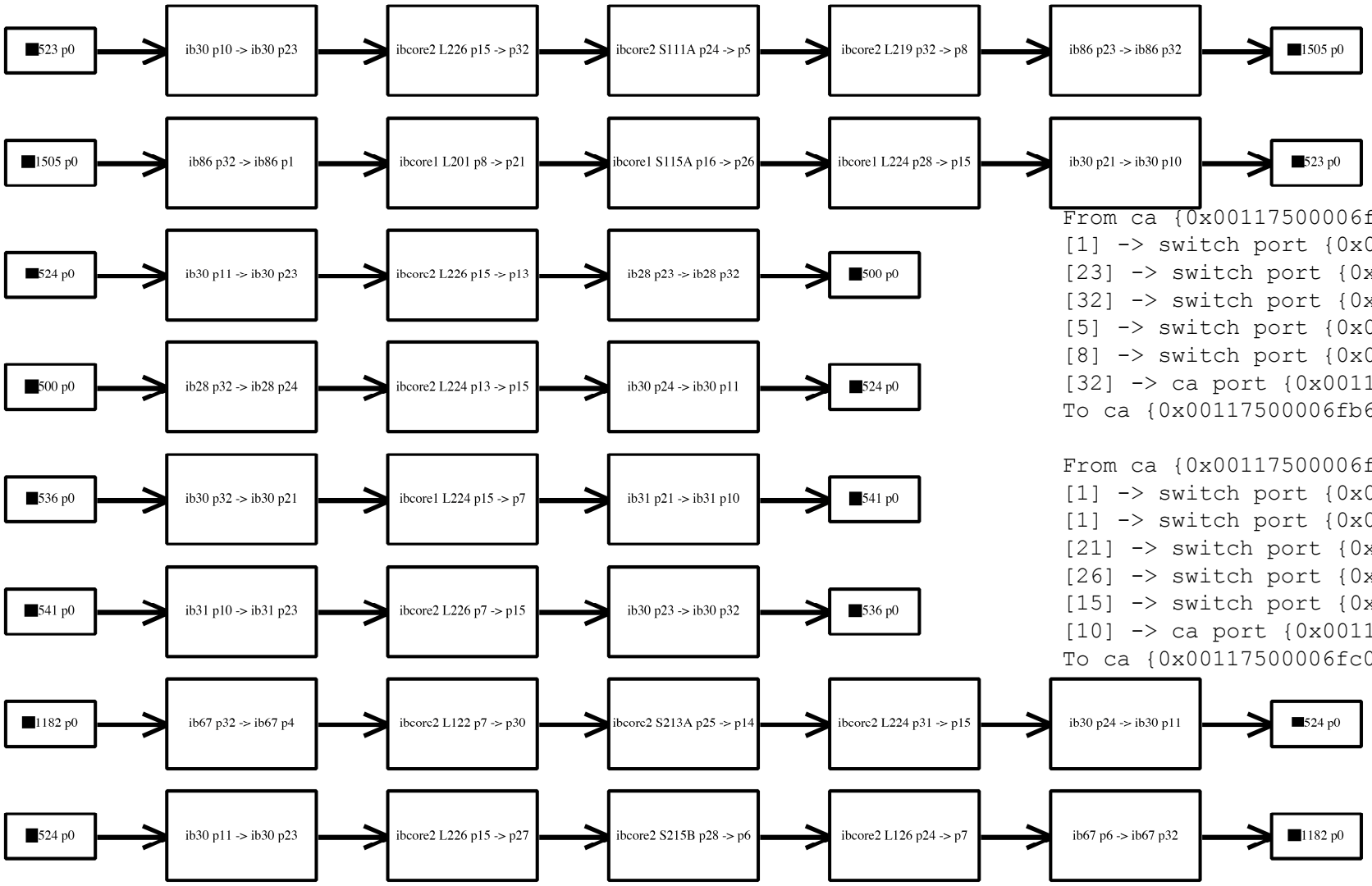
EXCEL .CSV INFO

- **Excel (on a Mac) Fails to Read in 320MB "Whole Day" — Can Read an Hour's Worth**

Entry #	Line #	Time	Time Delta	CompId	Port_Xmit_Wait Count	Delta	
1	4	1488092400		130024	2,082,299		
2	1266	1488092460	60	130024	792,893	1,289,406	reset prior
3	2531	1488092520	60	130024	800,462	7,569	
4	3790	1488092580	60	130024	1,141,028	340,566	
5	5054	1488092640	60	130024	1,174,117	33,089	
6	6317	1488092700	60	130024	1,535,149	361,032	
7	7579	1488092760	60	130024	1,544,015	8,866	
8	8838	1488092820	60	130024	1,982,079	438,064	
9	10172	1488092880	60	130024	1,990,948	8,869	
10	11434	1488092940	60	130024	2,147,879	156,931	
11	12654	1488093000	60	130024	2,580,935	433,056	
12	13959	1488093060	60	130024	344,518	2,236,417	reset prior

HANDY TOOL

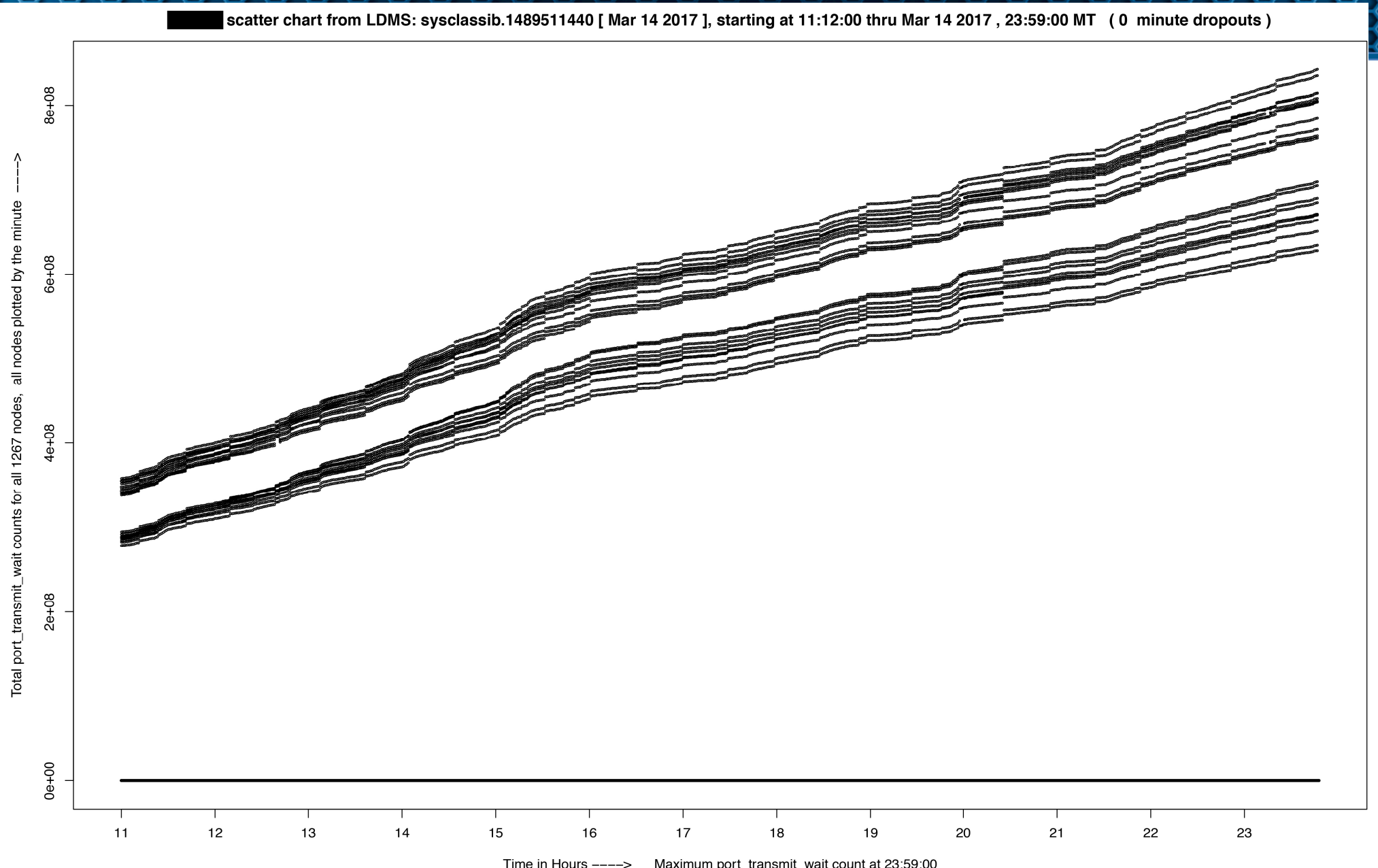
Created: Thu Aug 18 18:28:04 UTC 2016, by processing file Erik_data_clean.txt




```
From ca {0x00117500006fc026} portnum 1 lid 569-569 "ΔΔ523 qib0"
[1] -> switch port {0x00066a00e300679c}[10] lid 1175-1175 "ib30"
[23] -> switch port {0x00066a00ec003da7}[15] lid 2141-2141 "ibcore2 L226"
[32] -> switch port {0x00066a00eb0041d8}[24] lid 2153-2153 "ibcore2 S111A"
[5] -> switch port {0x00066a00ec003c3a}[32] lid 2122-2122 "ibcore2 L219"
[8] -> switch port {0x00066a00e30067f2}[23] lid 1938-1938 "ib86"
[32] -> ca port {0x00117500006fb6e2}[1] lid 1692-1692 "ΔΔ1505 qib0"
To ca {0x00117500006fb6e2} portnum 1 lid 1692-1692 "ΔΔ1505 qib0"

From ca {0x00117500006fb6e2} portnum 1 lid 1692-1692 "ΔΔ1505 qib0"
[1] -> switch port {0x00066a00e30067f2}[32] lid 1938-1938 "ib86"
[1] -> switch port {0x00066a00ec003c00}[8] lid 2055-2055 "ibcore1 L201"
[21] -> switch port {0x00066a00eb0041ec}[16] lid 2107-2107 "ibcore1 S115A"
[26] -> switch port {0x00066a00ec003d6c}[28] lid 2070-2070 "ibcore1 L224"
[15] -> switch port {0x00066a00e300679c}[21] lid 1175-1175 "ib30"
[10] -> ca port {0x00117500006fc026}[1] lid 569-569 "ΔΔ523 qib0"
To ca {0x00117500006fc026} portnum 1 lid 569-569 "ΔΔ523 qib0"
```

INTERESTING ....



INTERESTING ....