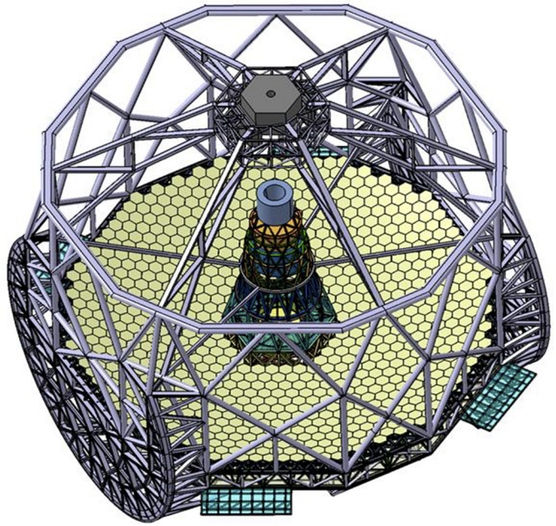
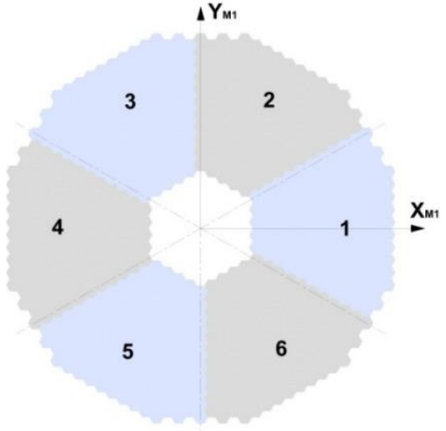
The ELT M1 Local Control Software: from Requirements to Implementation

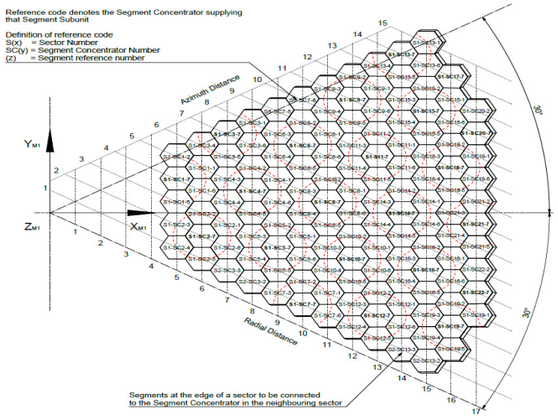
L. Andolfato, J. Argomedo, C. Diaz Cano, R. Frahm, T. Grudzien, N. Kornweibel, D. Ribeiro Gomes dos Santos, J. Sagatowski, C. M. Silva

M1 Local Control System - Overview

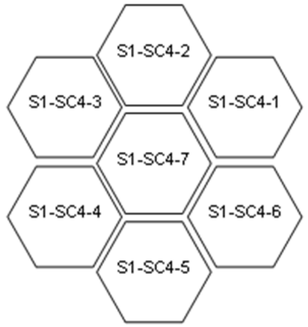


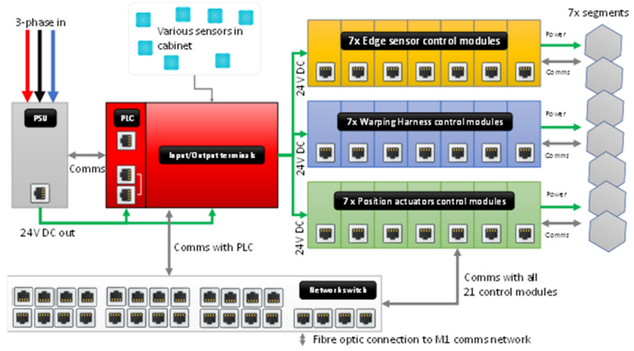


M1 = 39-m diameter, 798 hexagonal segments (1.4m each) grouped in 6 sectors.

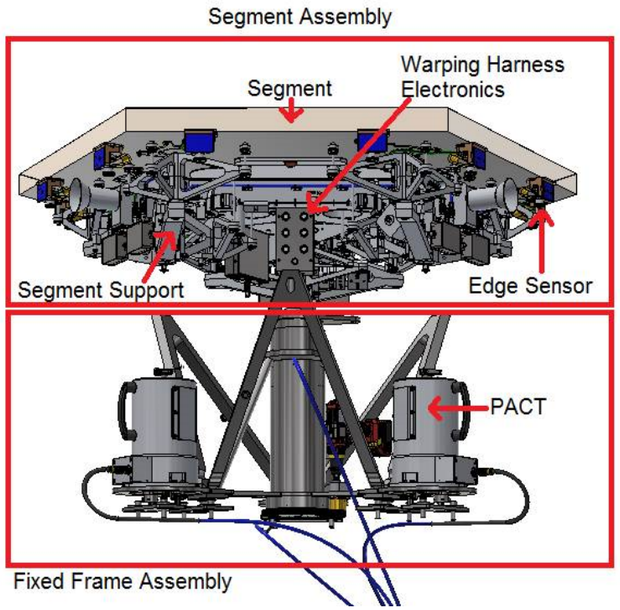


1 sector = 133 segments
(6 sector distributor cabinets)





1 flower = up to 7 segments
(132 segment concentrator cabinets)



1 segment is controlled by:

- ES controller (6 sensors piston/shear/gap)
- PACT controller (3 actuators + 3 sensors piston/tip/tilt)
- WH controller (9 actuators + 9 sensors)

M1LCS – Main SW Requirements

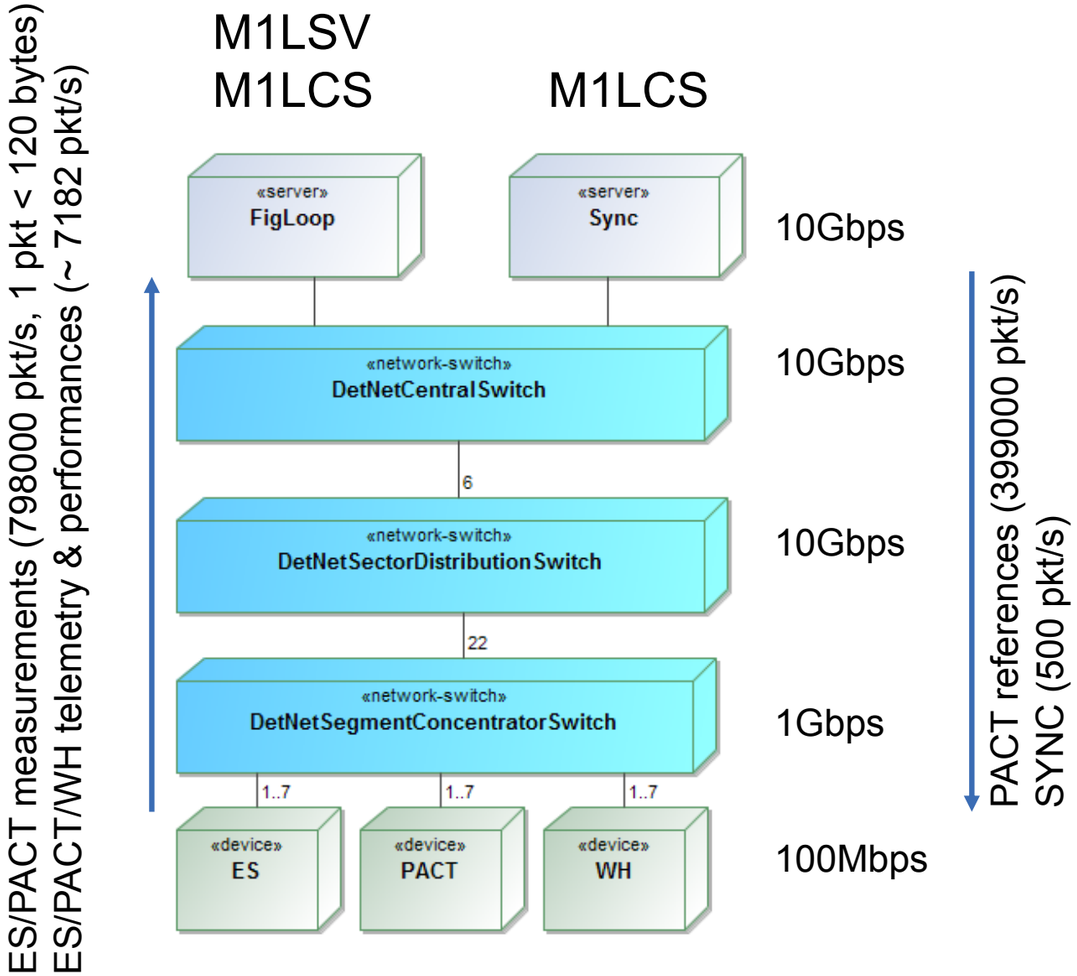
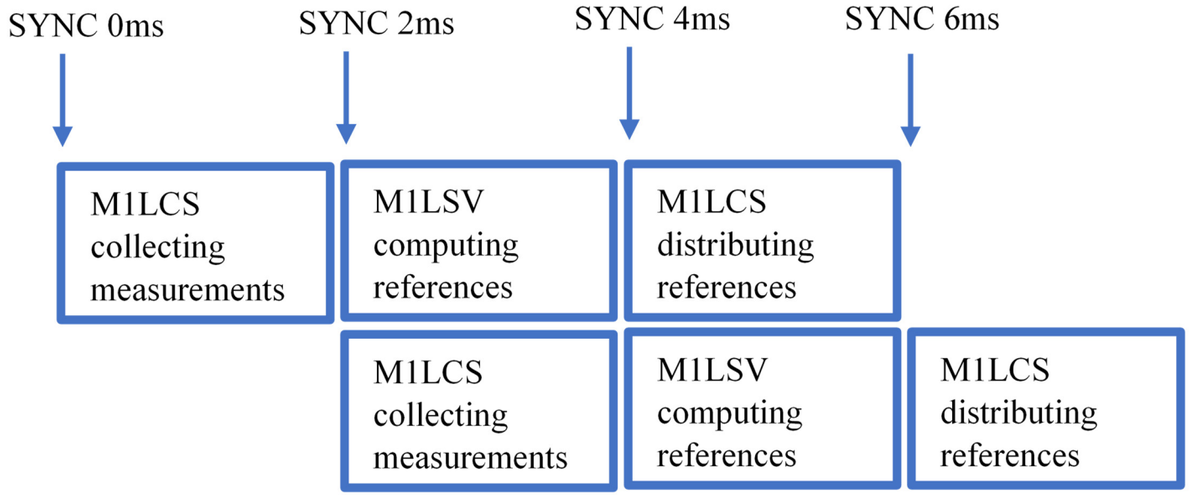
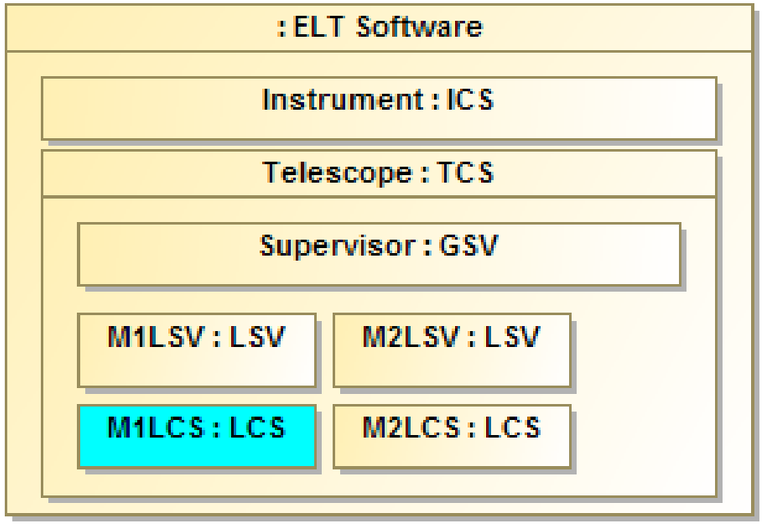


Figure Loop: obtain and maintain a given mirror optical quality for the duration of the observation.

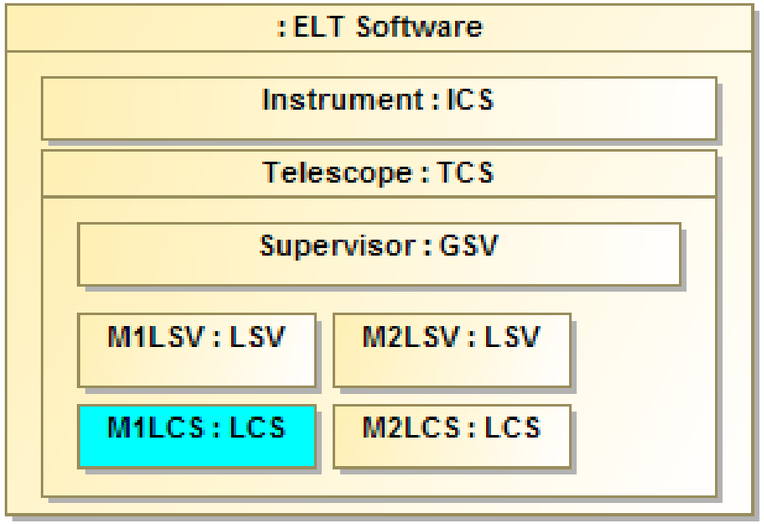
- Generate SYNC message every 2ms
- Collect ES/PACT measurements within 2ms (M1LCS)
- Compute new PACT references within 2ms (M1LSV)
- Apply new PACT references within 2ms (M1LCS)



M1LCS – SW Architecture

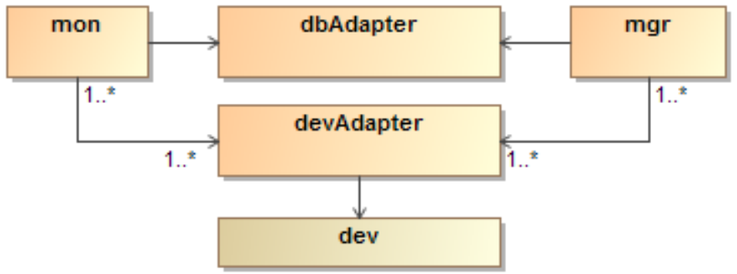


M1LCS – SW Architecture

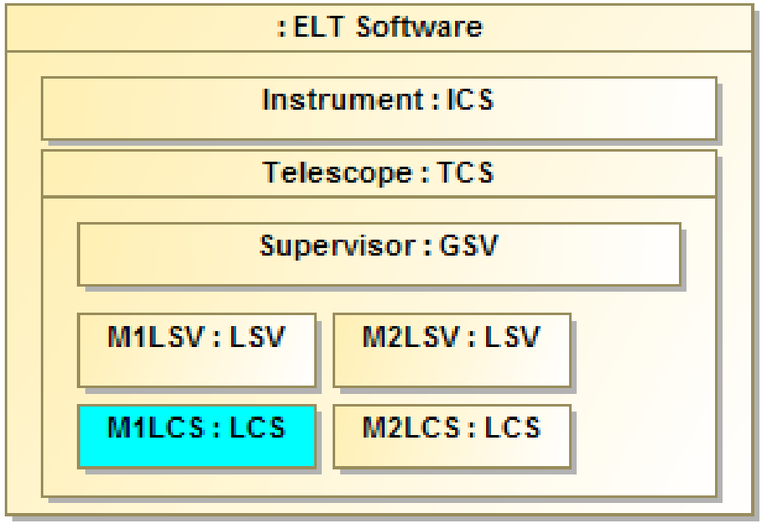


Adaptation of some JPL State Analysis patterns:

- Estimator/Controller/Adapter (Monitor/Manager/Adapter)
- FDIR based on Goal monitoring

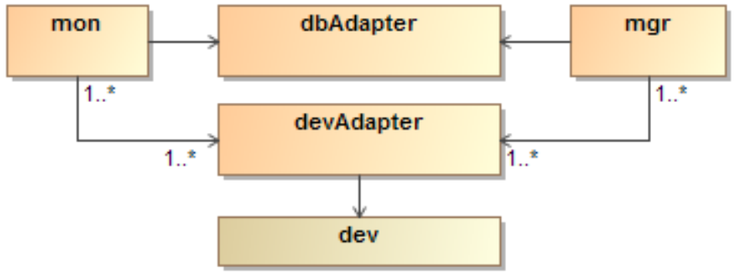


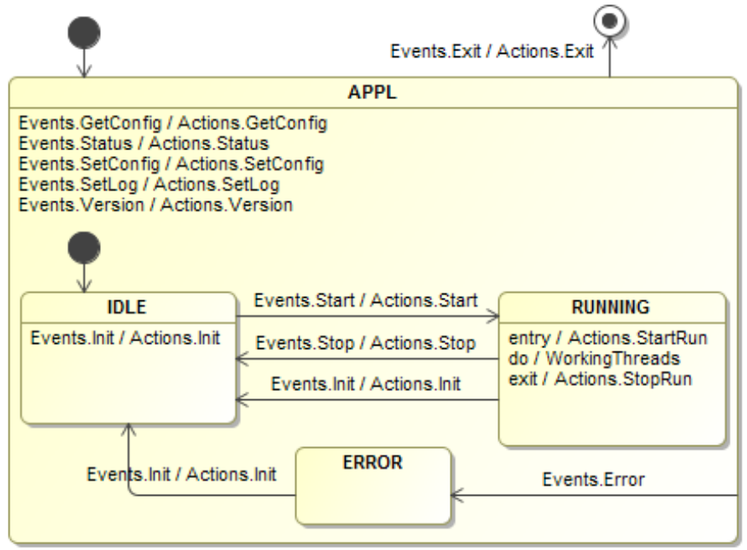
M1LCS – SW Architecture



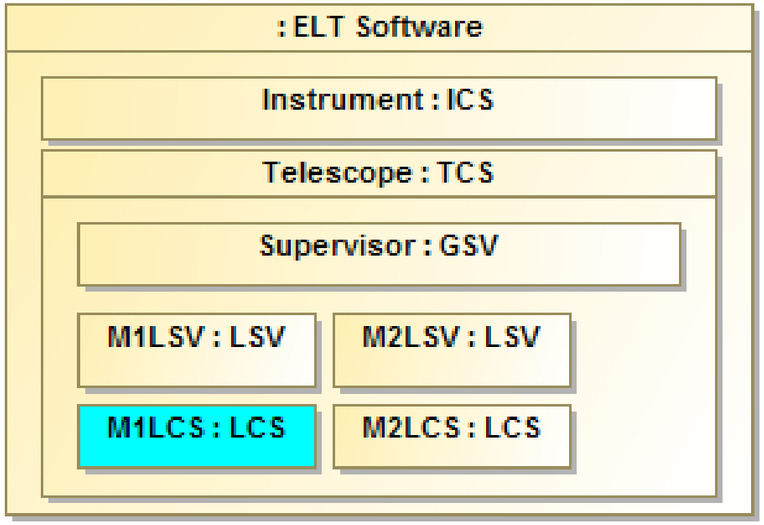
Adaptation of some JPL State Analysis patterns:

- Estimator/Controller/Adapter (Monitor/Manager/Adapter)
- FDIR based on Goal monitoring



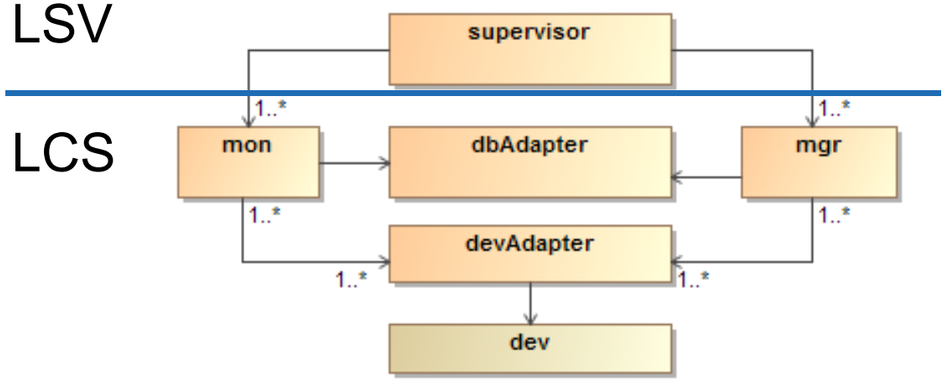


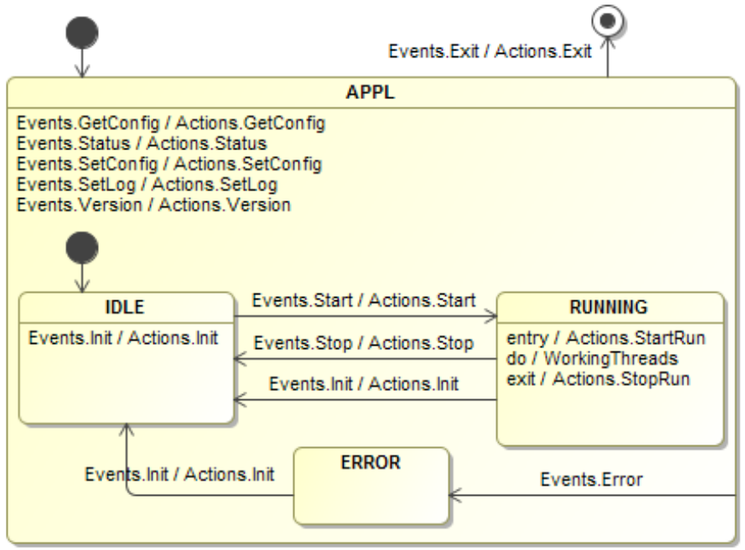
M1LCS – SW Architecture



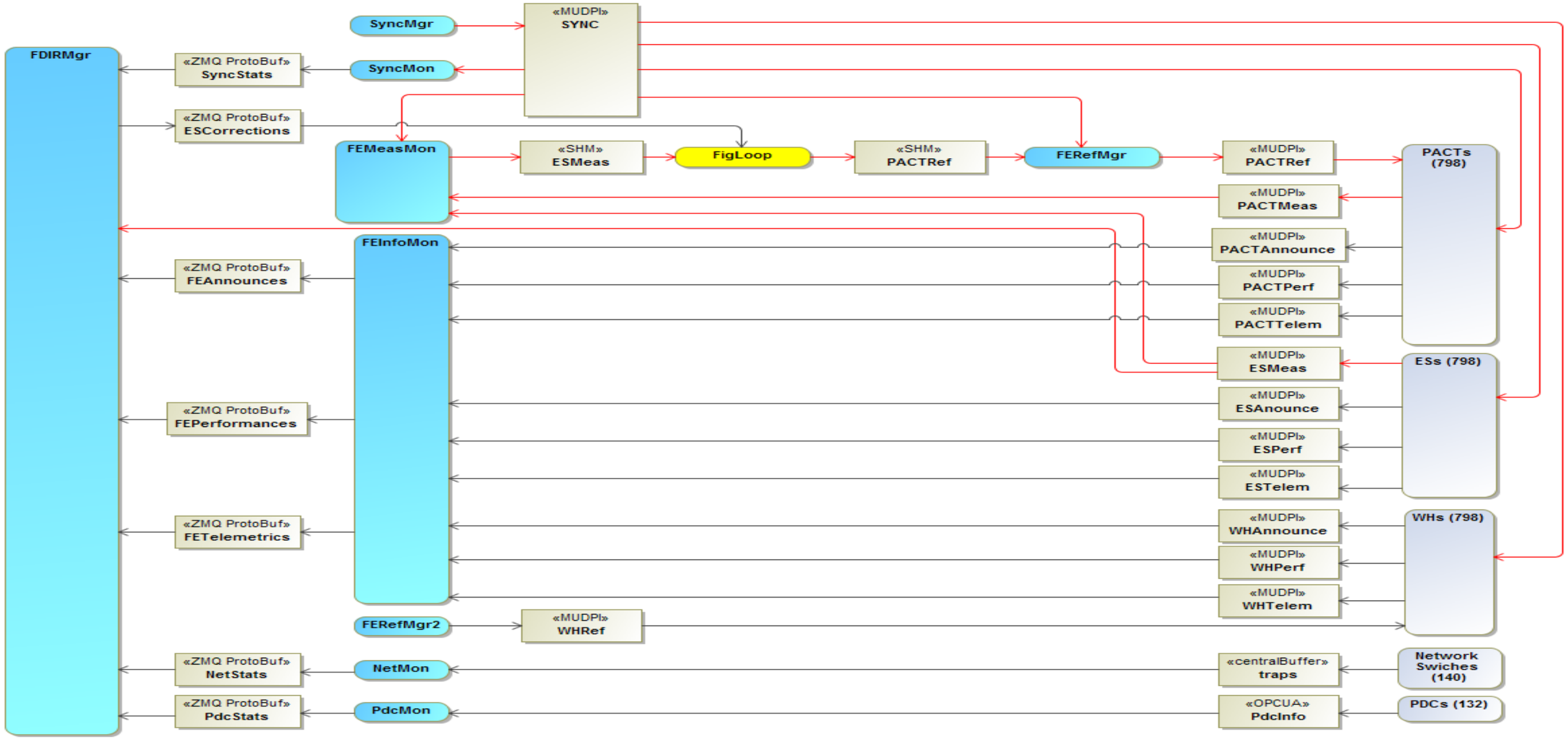
Adaptation of some JPL State Analysis patterns:

- Estimator/Controller/Adapter (Monitor/Manager/Adapter)
- FDIR based on Goal monitoring

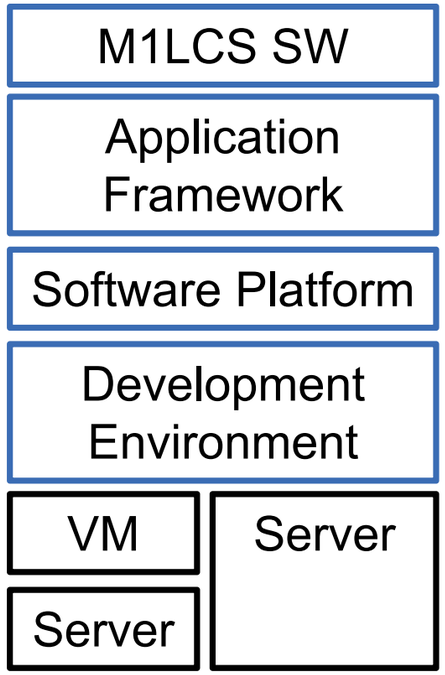




M1LCS - Data Flow (500Hz in red)



M1LCS – SW Implementation

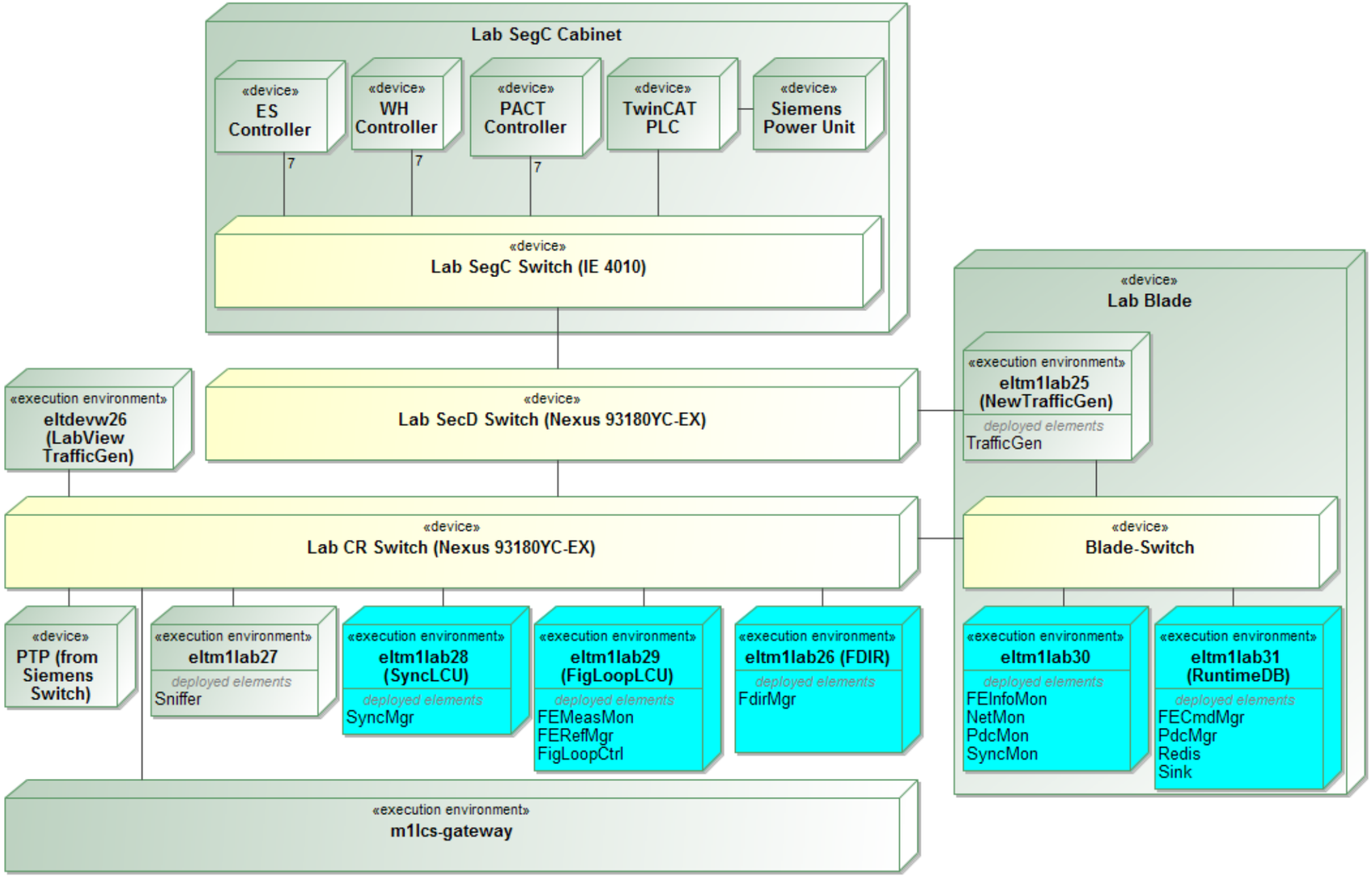


Development Environment	
OS	Linux CentOS RT patch (CERN dist.)
Languages	C++/C, Python
Building	Waf (+ ESO wtools)
GUI	Qt, PySide2
Unit Tests	Google Tests
Int. Tests	Robot Framework
Deployment	Nomad/Consul
CI	Jenkins
Quality	Cpplint, cppcheck, valgrind

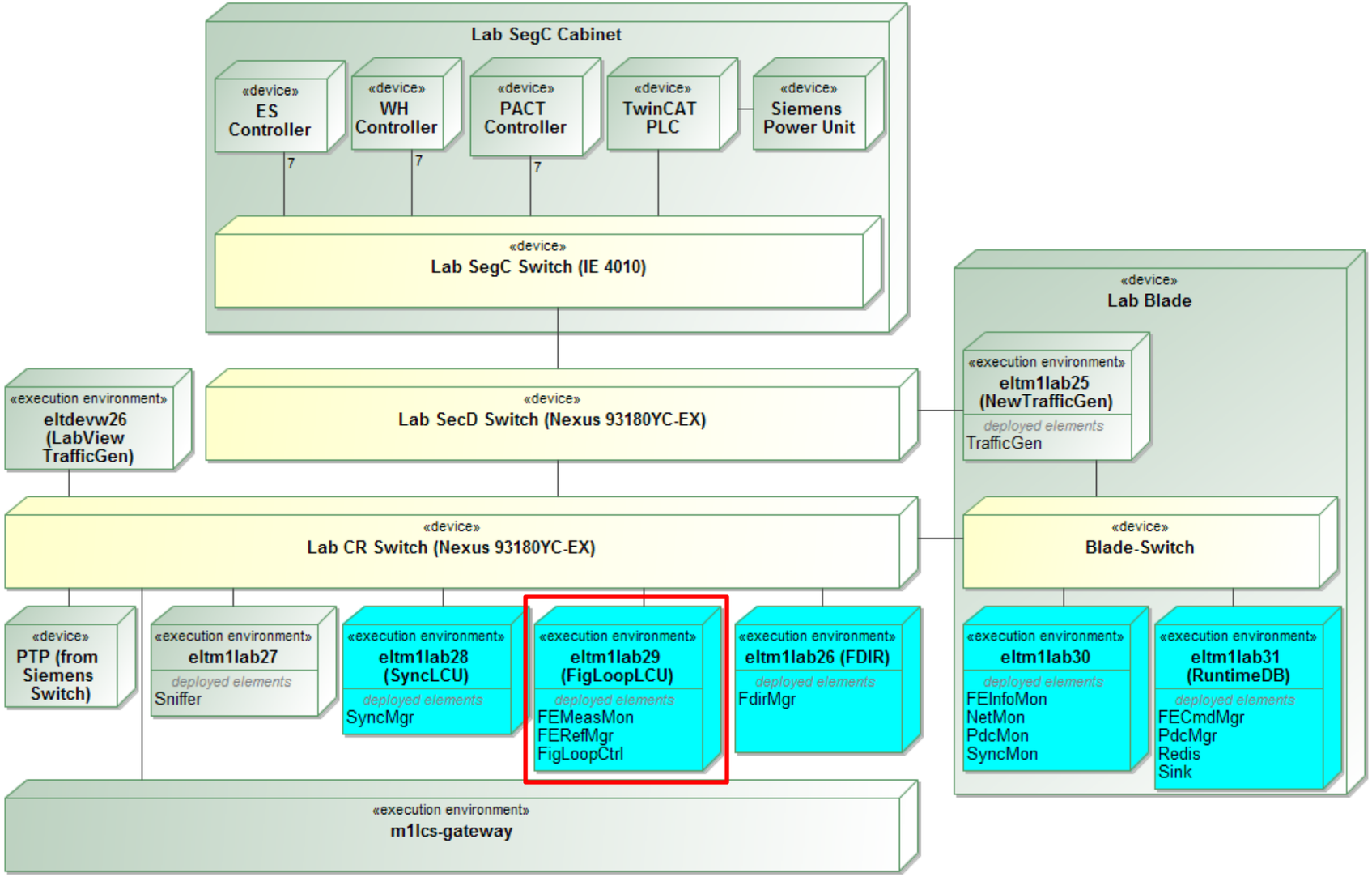
Application Framework (RAD)	
Event Loop	Boost ASIO + AZMQ
State Machine Engine	SCXML interpreter: scxml4cpp (ESO)
Code generation	Event DSL / codegen (ESO) SysML Statecharts / COMODO (ESO) Templates / Cookiescutter

Software Platform (Temporary)	
Middleware	ZMQ, MUDPI (UDP), OPC-UA (open62541); [ZMQ ser/deser: Google ProtoBuf]
Configuration	File based (YAML) + Redis
In-memory DB	Redis (hiredis)
Error Handling	Exception
Logging	EasyLogging
Alarm	Based on Redis

M1LCS – Lab Deployment



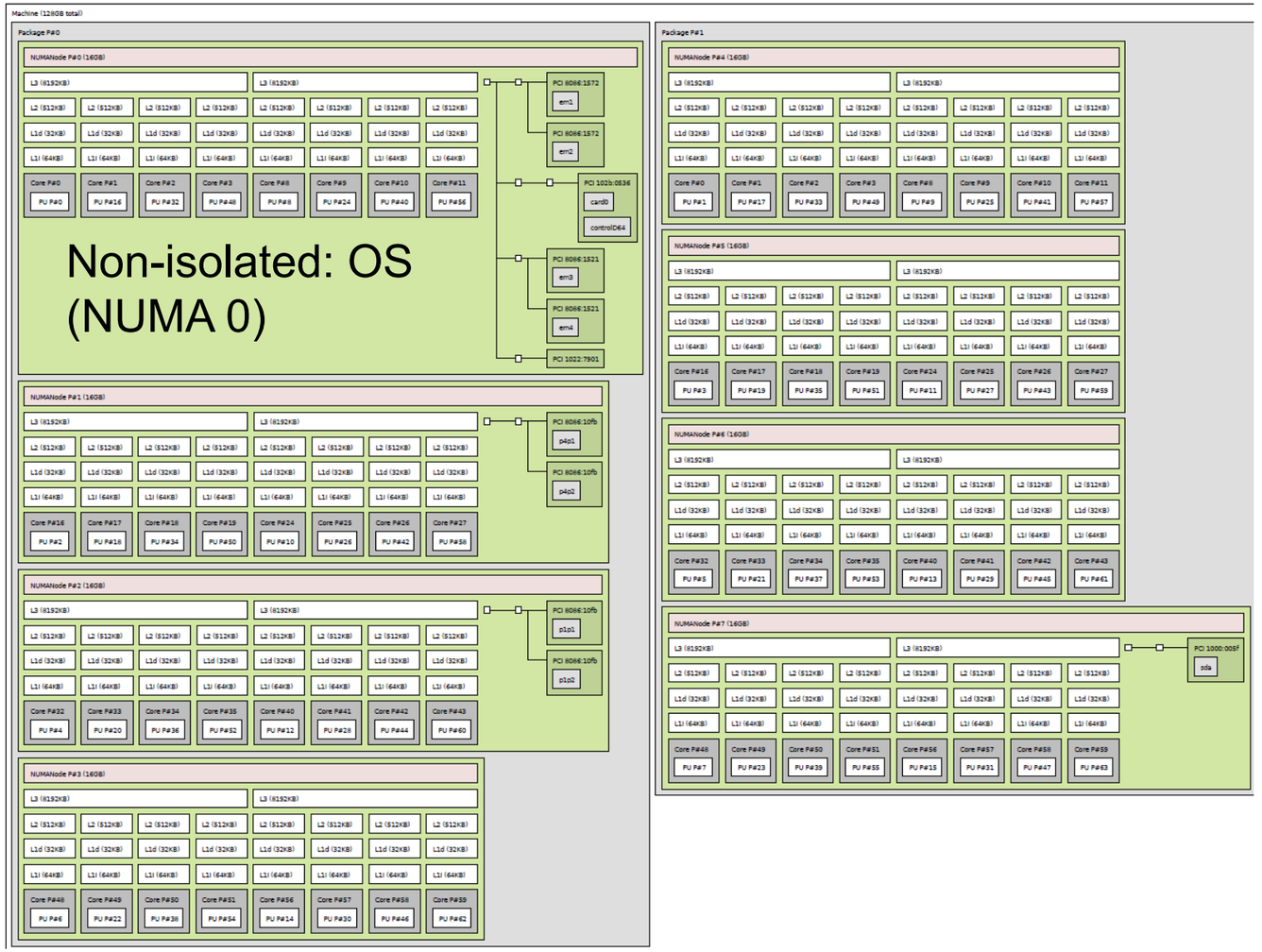
M1LCS – Lab Deployment



M1LCS/LSV – Figure Loop on AMD Epyc

1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4+4 cores (2 core-cortex)
- 8+8MB L3
- 1+1 memory channels

Die -> NUMA node -> PCI



>lstopo
>numactl -H

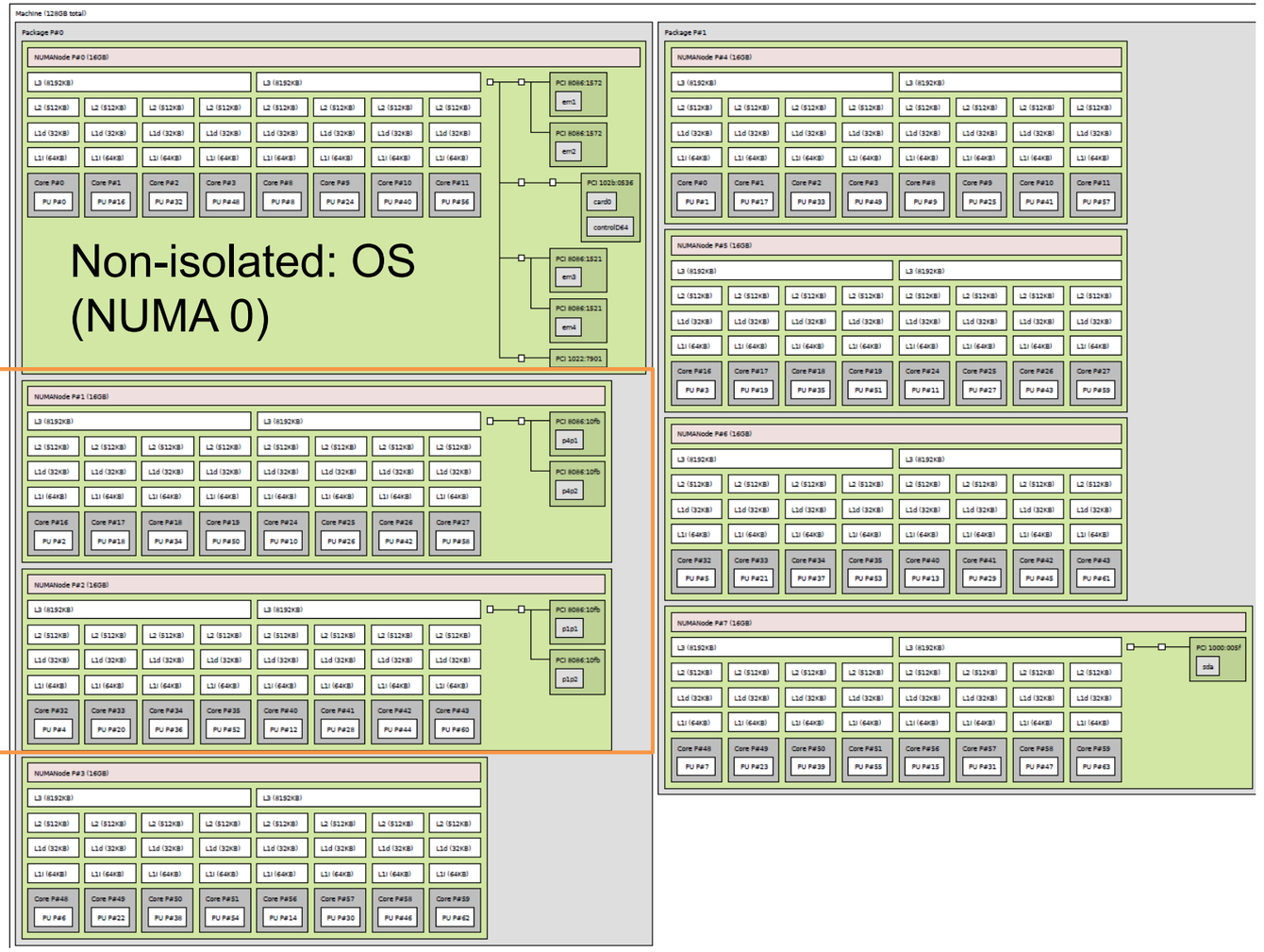
CPU-0 (M1LCS) **CPU-1 (M1LSV)**

M1LCS/LSV – Figure Loop on AMD Epyc

1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4+4 cores (2 core-cortex)
- 8+8MB L3
- 1+1 memory channels

Die -> NUMA node -> PCI

FEMeasMon
6+2 threads to acquire ES+PACT meas.
(NUMA 1,2 -> 2 PCI slots/NICs)



>lstopo
>numactl -H

CPU-0 (M1LCS) **CPU-1 (M1LSV)**

M1LCS/LSV – Figure Loop on AMD Epyc

1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4+4 cores (2 core-cortex)
- 8+8MB L3
- 1+1 memory channels

Die -> NUMA node -> PCI

FEMeasMon
6+2 threads to acquire ES+PACT meas.
(NUMA 1,2 -> 2 PCI slots/NICs)

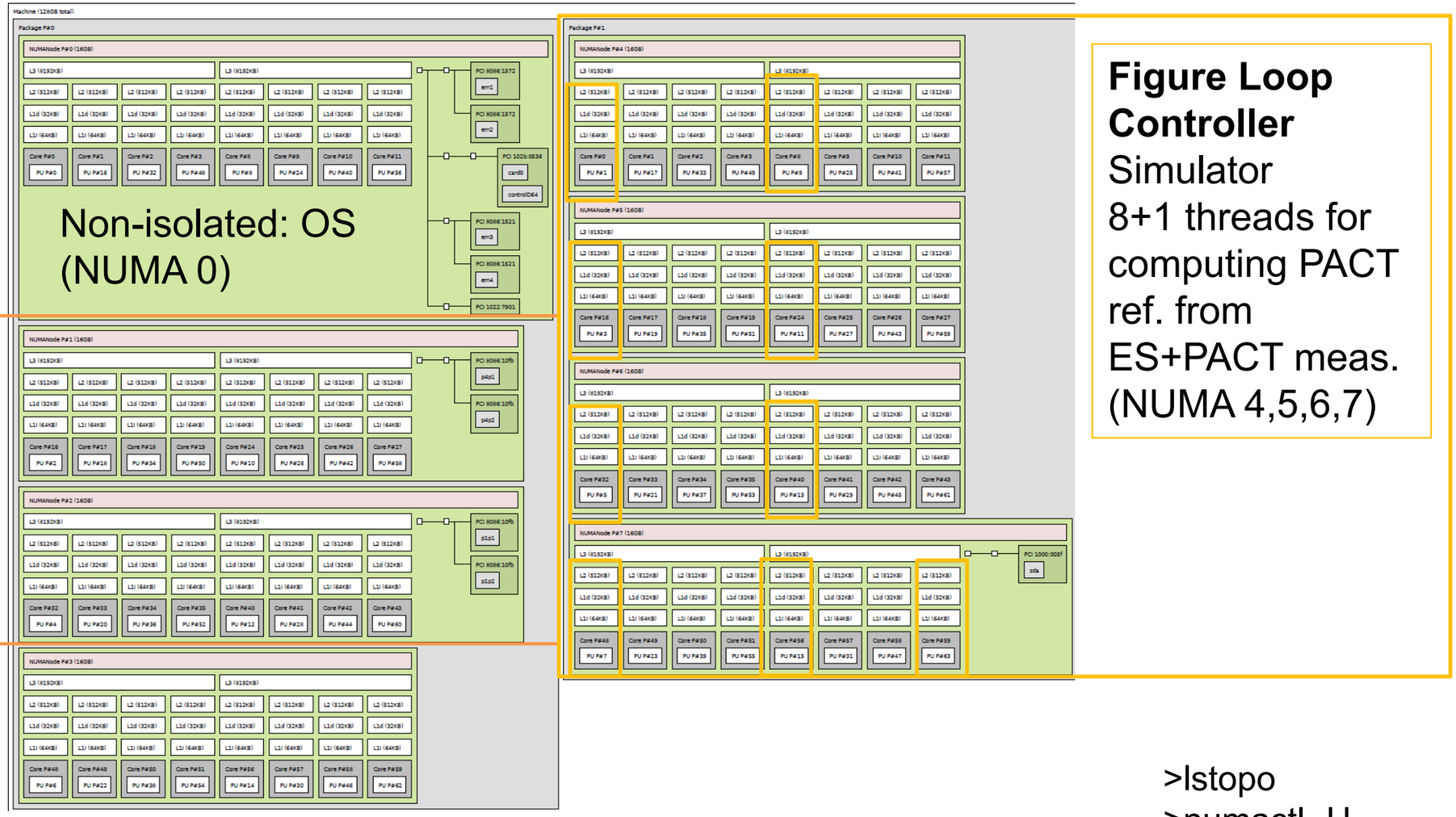


Figure Loop Controller
Simulator
8+1 threads for computing PACT ref. from ES+PACT meas.
(NUMA 4,5,6,7)

>lstopo
>numactl -H

CPU-0 (M1LCS) **CPU-1 (M1LSV)**

M1LCS/LSV – Figure Loop on AMD Epyc

1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4+4 cores (2 core-cortex)
- 8+8MB L3
- 1+1 memory channels

Die -> NUMA node -> PCI

FEMeasMon
6+2 threads to acquire ES+PACT meas.
(NUMA 1,2 -> 2 PCI slots/NICs)

FERefMgr
4+2 threads to send PACT ref. (NUMA 3)

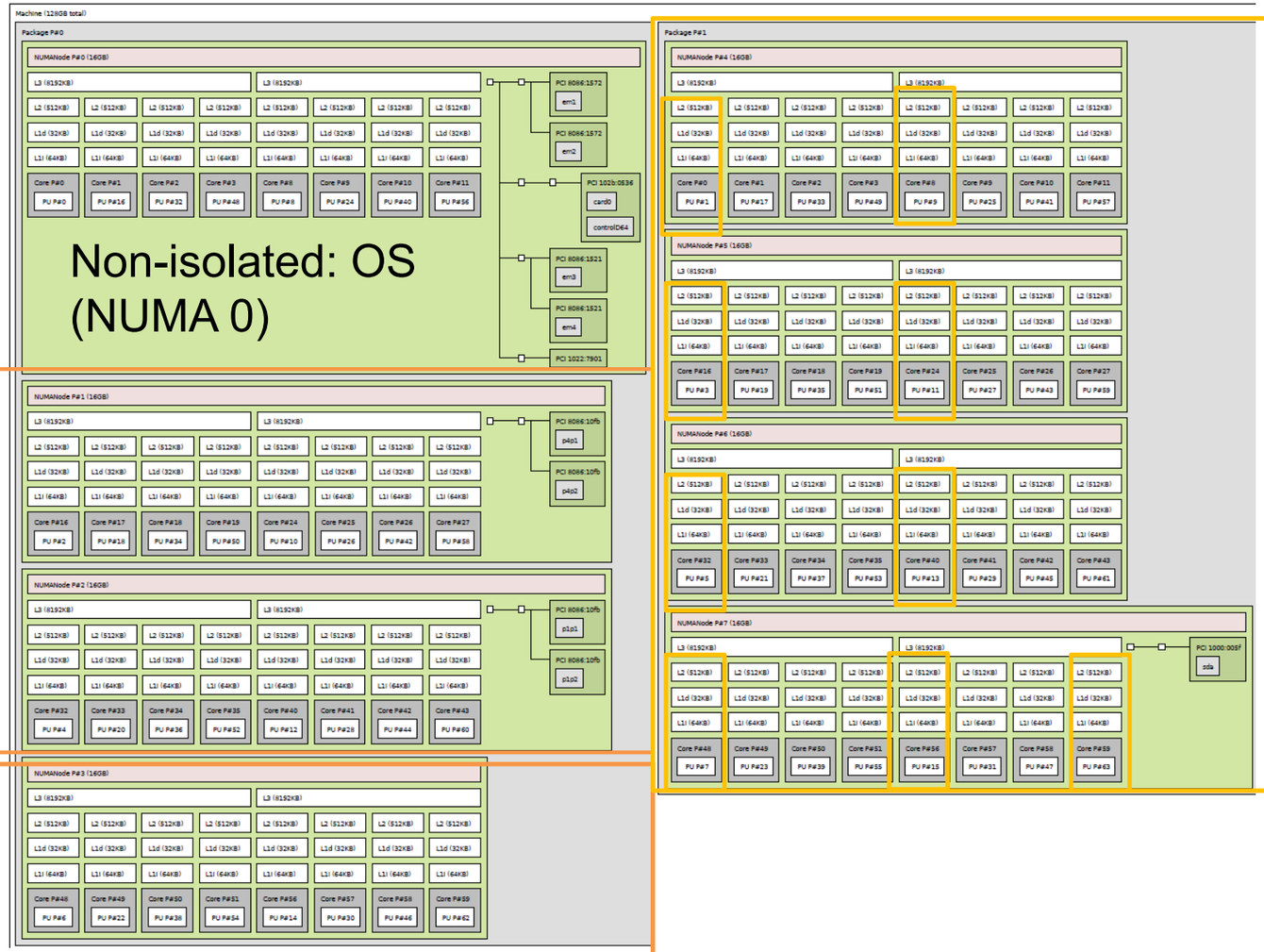


Figure Loop Controller
Simulator
8+1 threads for computing PACT ref. from ES+PACT meas.
(NUMA 4,5,6,7)

>lstopo
>numactl -H

CPU-0 (M1LCS) **CPU-1 (M1LSV)**

M1LCS/LSV – Figure Loop on AMD Epyc

1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4+4 cores (2 core-cortex)
- 8+8MB L3
- 1+1 memory channels

Die -> NUMA node -> PCI

FEMeasMon
6+2 threads to acquire ES+PACT meas.
(NUMA 1,2 -> 2 PCI slots/NICs)

FERefMgr
4+2 threads to send PACT ref. (NUMA 3)

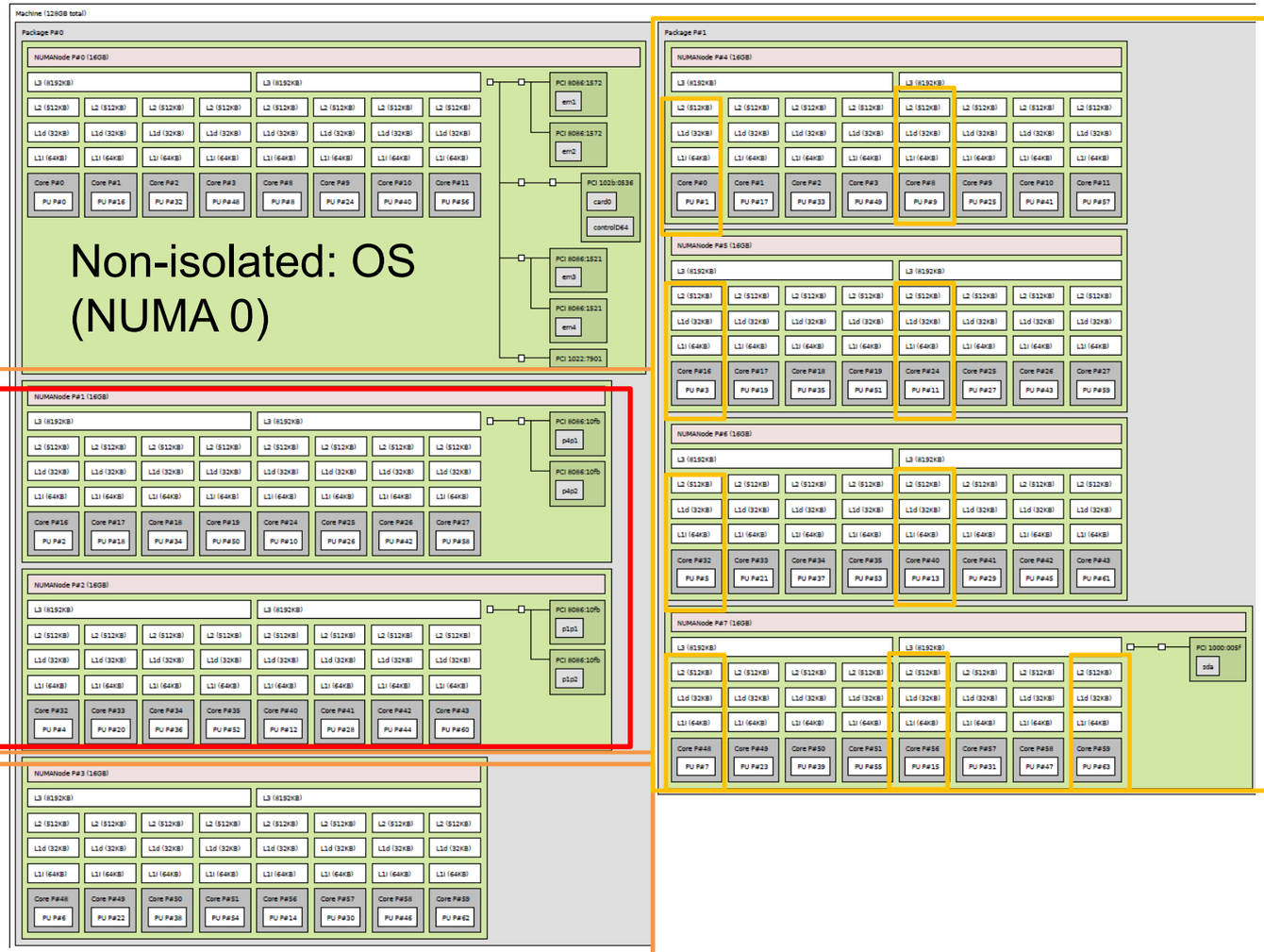
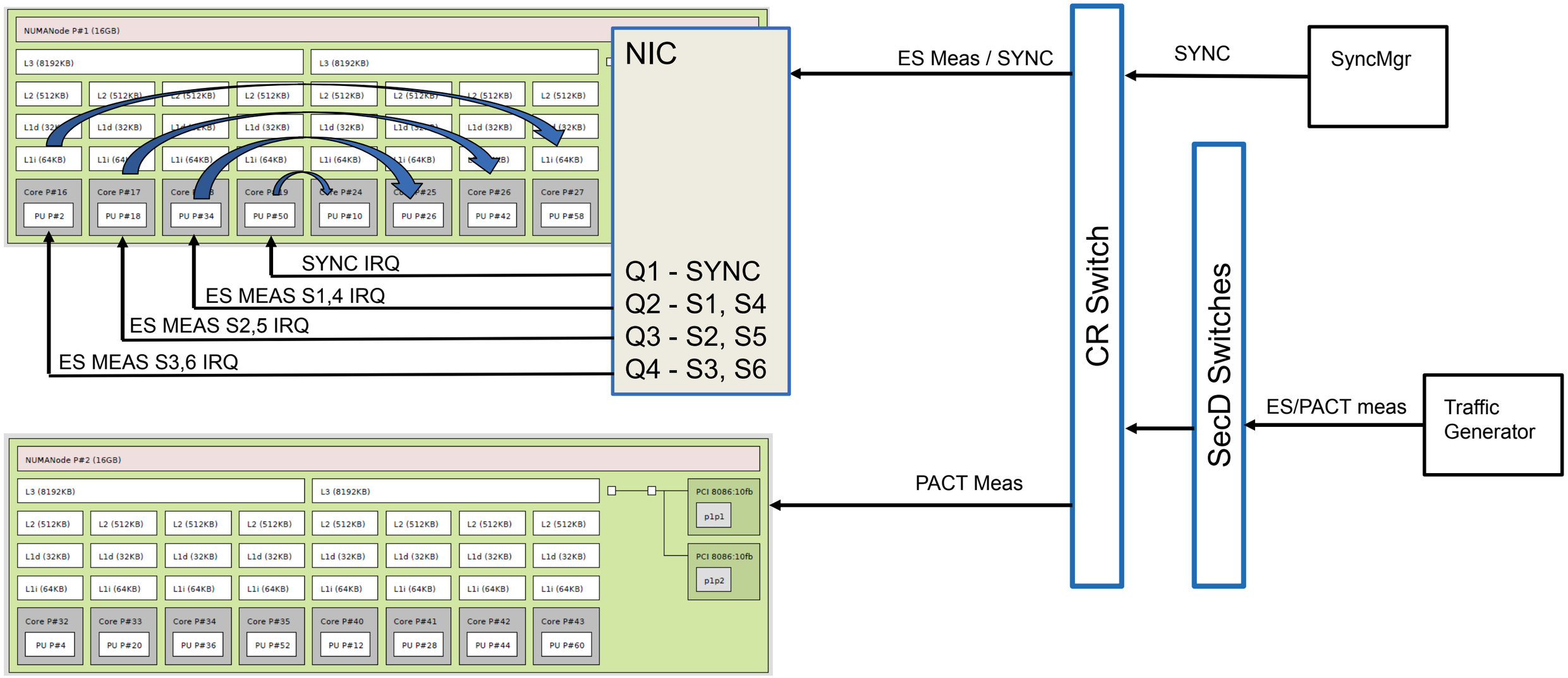


Figure Loop Controller
Simulator
8+1 threads for computing PACT ref. from ES+PACT meas.
(NUMA 4,5,6,7)

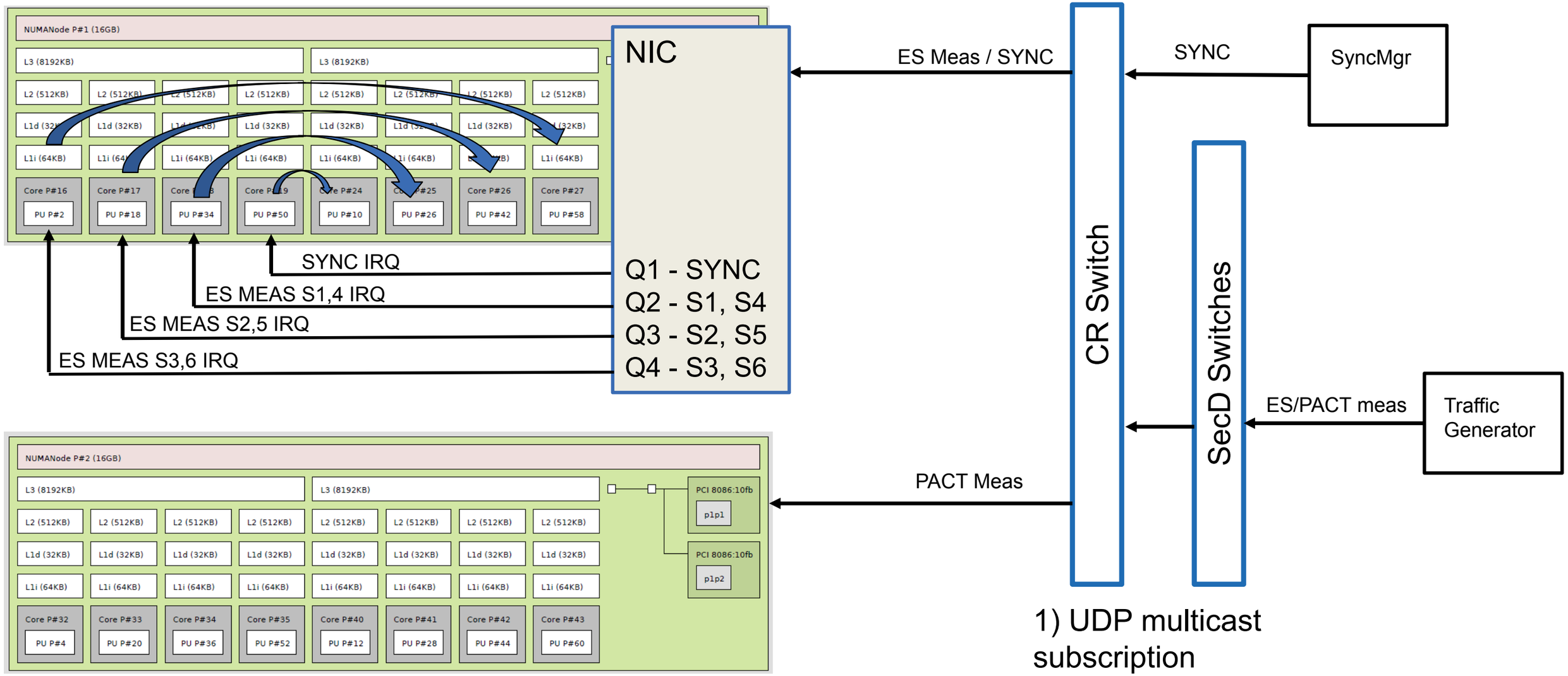
>lstopo
>numactl -H

CPU-0 (M1LCS) **CPU-1 (M1LSV)**

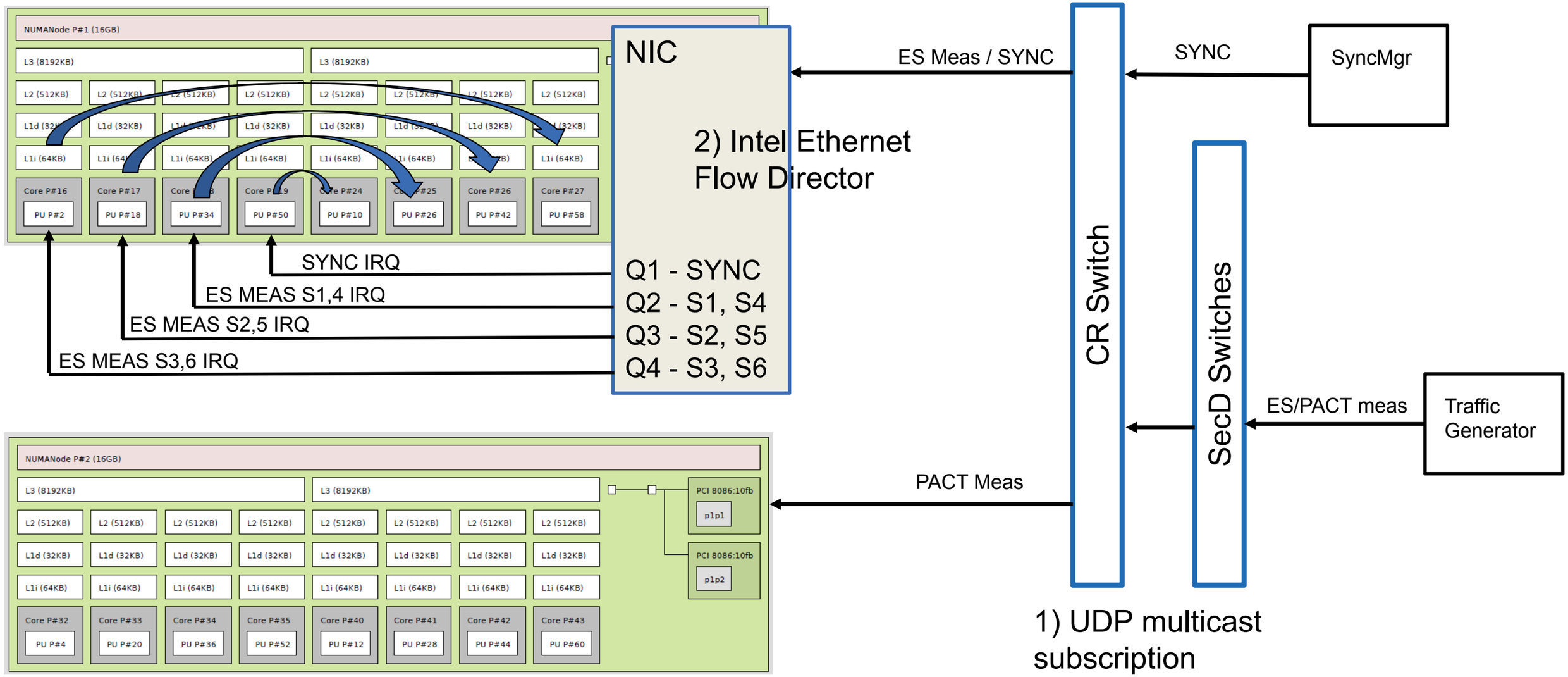
M1LCS – Measurements Packet Flow



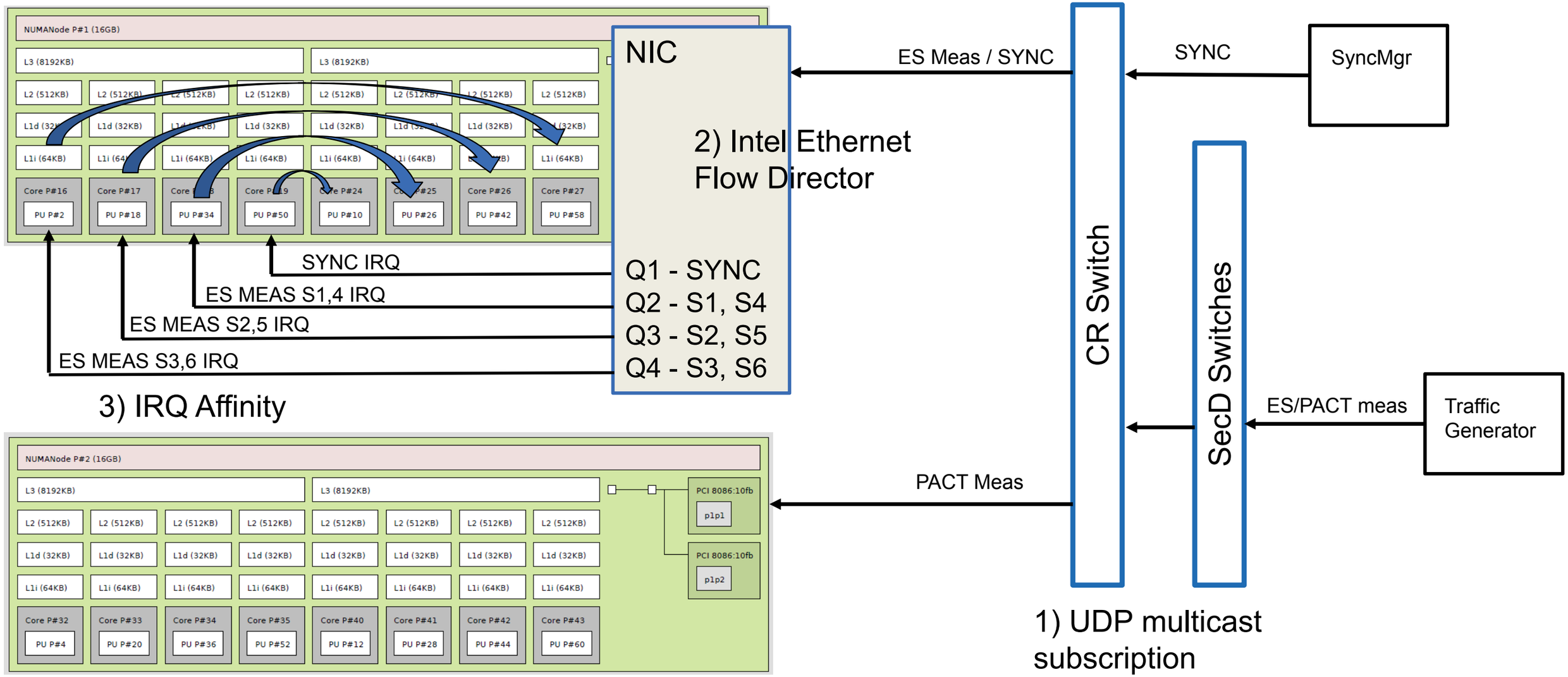
M1LCS – Measurements Packet Flow



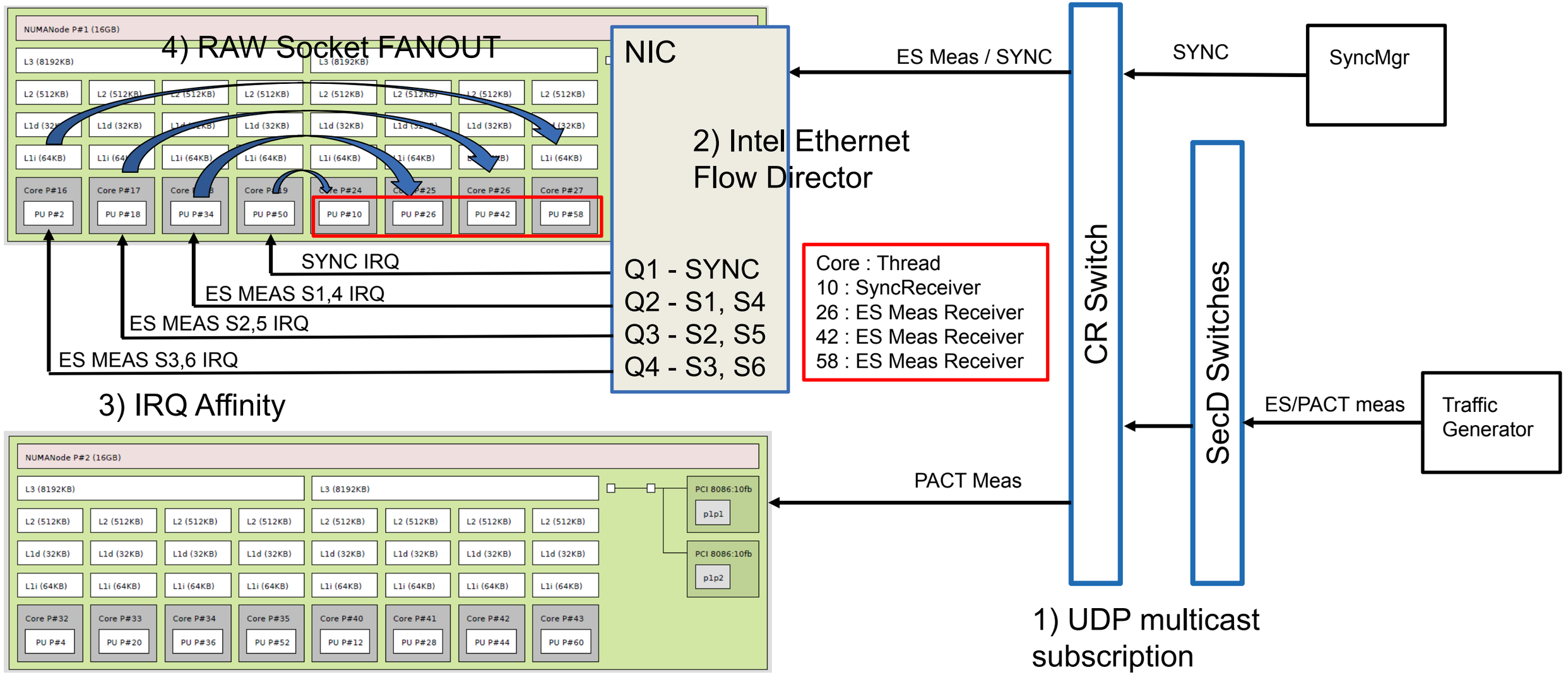
M1LCS – Measurements Packet Flow



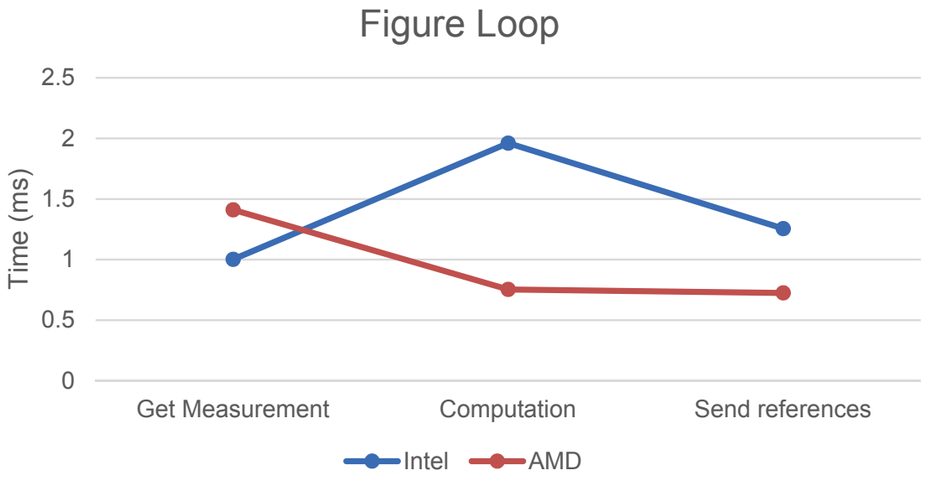
M1LCS – Measurements Packet Flow



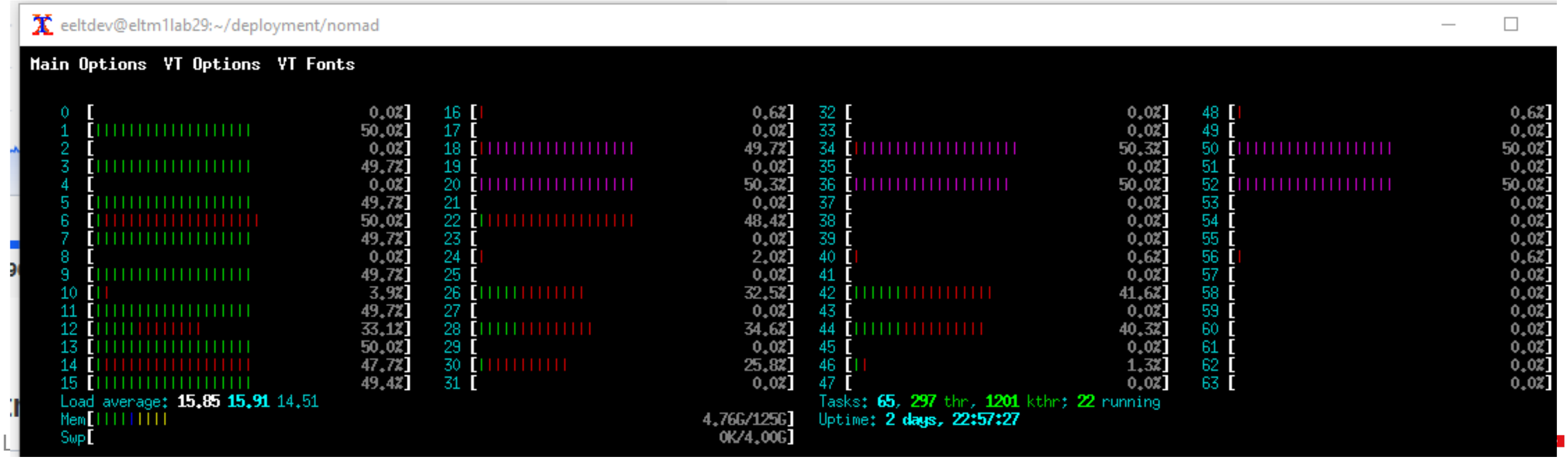
M1LCS – Measurements Packet Flow



M1LCS – Figure Loop Results



m1FEMeasMon.ctr.update.cycle_time	0.001242	< 0.002s
m1FEMeasMon.ctr.update.time_last_packet	0.001139	
m1FigLoop.ctr.update.cycle_time	0.000685	< 0.002s
m1FERefMgr.ctr.update.cycle_time	0.000537	< 0.002s



Questions?