Performance Interfaces for PCIe Topologies


Alexandre Piveteau


George Candea

DSLAB

Problem: understanding PCIe performance behavior is difficult

Why PCIe is hard to reason about:

- **PCIe is a packet protocol:** every transfer carries header and protocol overhead
- **Reads are split transactions:** requests and completions travel separately
- **Many requests can be in flight:** devices track tens to hundreds of outstanding transactions.
- **Throughput and latency depend on request size, concurrency and the bus topology.**
- **Measurements show what happens, but not why.**

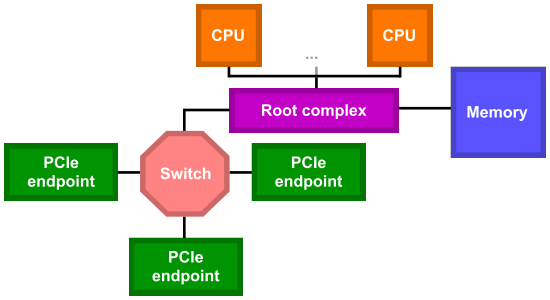
We want a performance interface which is:

- **Predictive**
  → enables throughput/latency/bottleneck analysis
- **Composable**
  → combines across system components
- **Resolution-adjustable**
  → can trade accuracy for simplicity
- **Minimal**
  → exposes only performance-relevant elements

From PCIe hardware/software to a performance model

We model PCIe buffering, delays and credit-based flow control using Latency Petri Nets (LPNs).

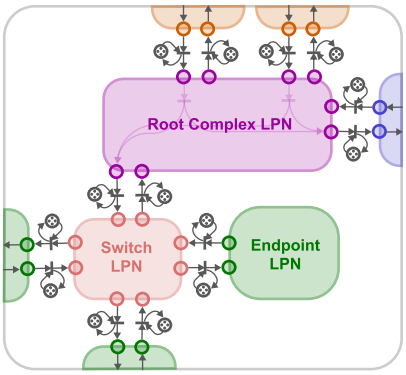
PCIe topology



Performance equivalent

Developers compose existing, high-level LPNs to model their system

Corresponding PCIe LPN



LPNs offer a formal framework to reason about PCIe behavior

Steady-state flow equations (Linear Programming)

Latency Petri Net: $N = (P, T, I, O, E)$

Edge flows: $(u,v) \in F \triangleq \{(p,t) \mid p \in I(t)\} \cup \{(t,p) \mid p \in O(t)\} \quad \varphi((u,v)) \in \mathbb{R}_{\geq 0}$

Constraints:

For each $t \in T, x_t \in \mathbb{R}_{\geq 0}$*:*

$$\sum_{t|(t,p)\in F} \varphi((t,p)) = \sum_{t|(p,t)\in F} \varphi((p,t)) \qquad \begin{aligned} \varphi((p,t)) &= W_I(t,p)\, x_t \quad \text{for all } p \in I(t), \\ \varphi((t,p)) &= W_O(t,p)\, x_t \quad \text{for all } p \in O(t). \end{aligned} \qquad \varphi((p,t)) \leq \frac{K_i W_I(t,p)}{D_i}$$

Place conservation | Transition consistency | Cycle delay bounds

We can maximise the flow to obtain upper bounds on throughput!

Recurring markings and simulation

LPN State: $S = (M, A, \theta)$

$M : P \to \text{Tok}^* \qquad A : P \to (\mathbb{N} \times \text{Tok})^* \qquad \theta \in \mathbb{N}$

Recurrence search over sim. trace

```
Procedure FIRST_RECURRENCE_OR_NULL(M0, A0)
    SEEN <- empty
    for each (M, A) in simulate(M0, A0):
        if (M, A) in SEEN:
            return (M, A)
        SEEN <- SEEN union {(M, A)}
    return null
```

We can identify recurring patterns to estimate throughput!

Latency Petri Nets can capture the mechanisms that shape PCIe performance in a form that is both analyzable and compact.

Interested in related topics? Talk to us or visit **dslab.epfl.ch!**