Carving Software-Defined Networks for Scientific Applications with SPATEN

Celio Trois[1], Luis Bona[1], Marcos Didonet del Fabro[1]
email: {ctrois, bona, didonet}@inf.ufpr.br

Magnos Martinello[2]
email: magnos@inf.ufes.br

Introduction

Scientific Applications (SciApps)

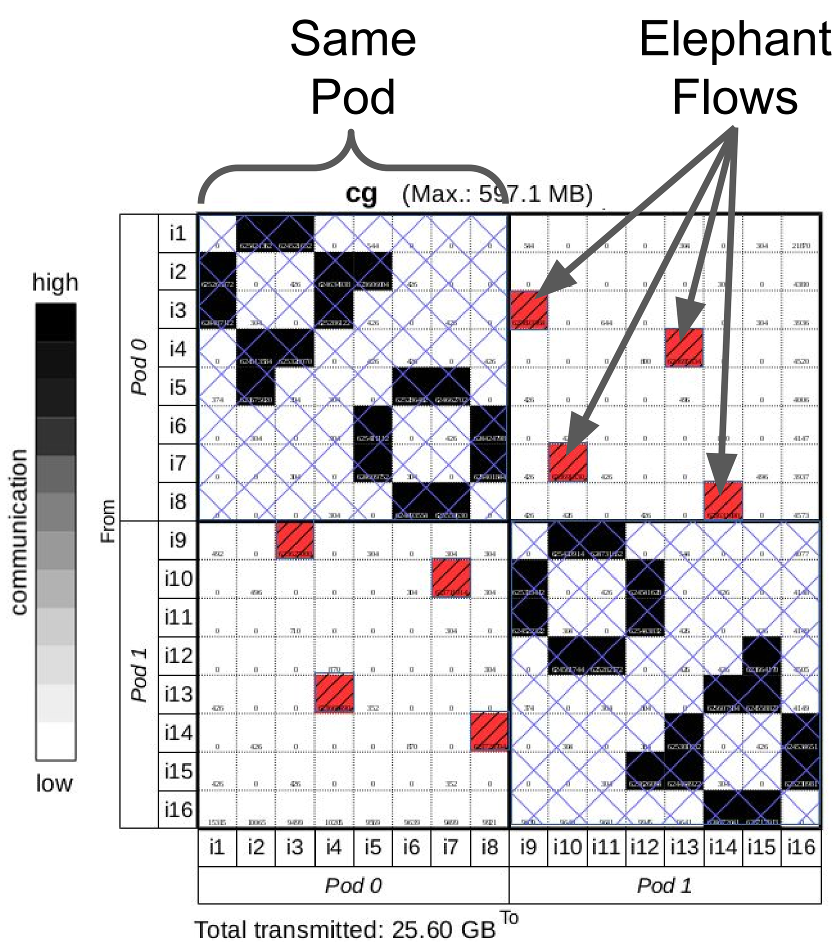
- Present **well-behaved communication patterns** [1], transmitting a similar amount of data across same nodes.
- Performance highly dependent on the nodes interconnection bandwidth

Software-Defined Networking

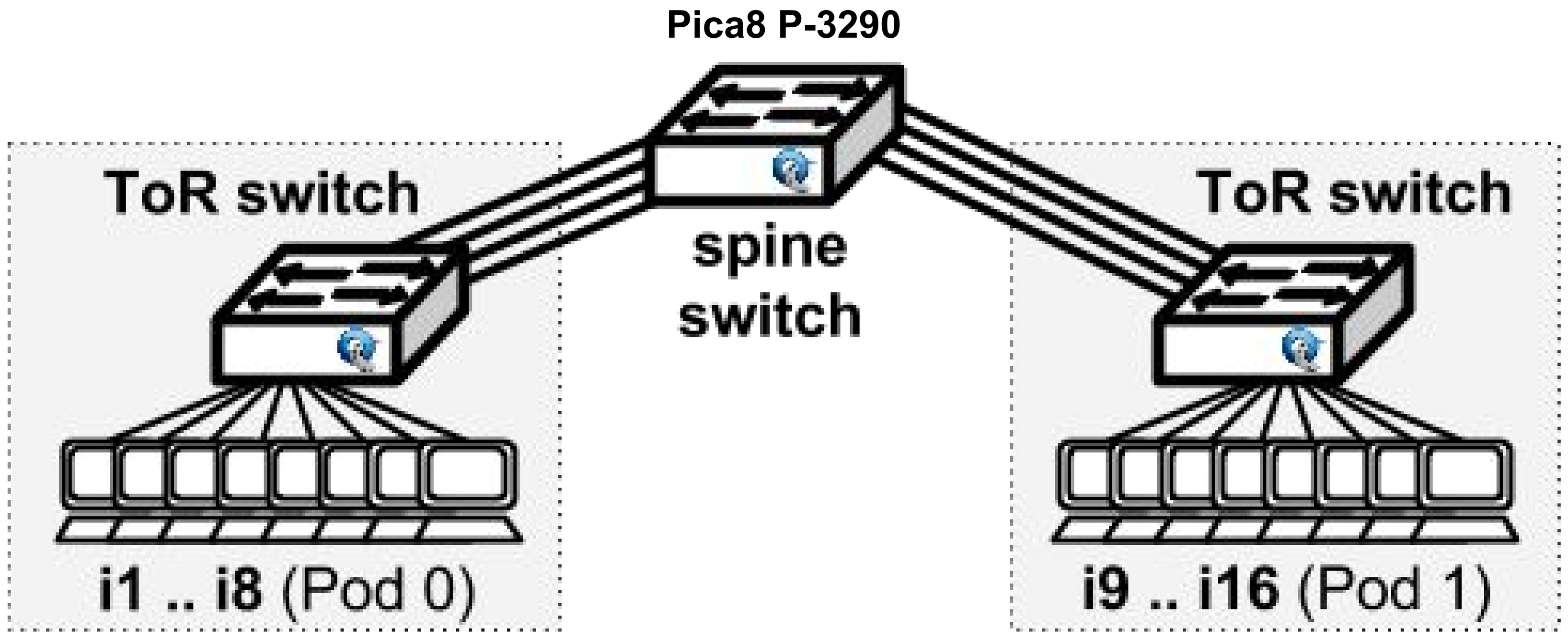
- Supports new possibilities for network management
- **Network behavior can be modified on-the-fly** according to user requirements
- Introduces **new issues**
  - Populating switch forwarding tables is time-consuming
  - Huge number of flow table entries burst TCAM size

SPATEN

- Exploits **S**patial **PAT**terns to **E**nhance **N**etwork for SciApps
- Previously stores spatial behaviors
- Identifies the elephant flows [2]
- Reduce the number of flow table entries by grouping matching rules by destination address [3]
- Uses Dijkstra's weighted shortest-path algorithm [4] for placing the inter-PoD elephant flows
- Proactively installs the forwarding rules on the switches



Real Testbed Topology



1. Univ. Federal do Paraná (UFPR) - Curitiba - Brazil
2. Univ. Federal do Espírito Santo (UFES) - Espirito Santo - Brazil

Evaluation

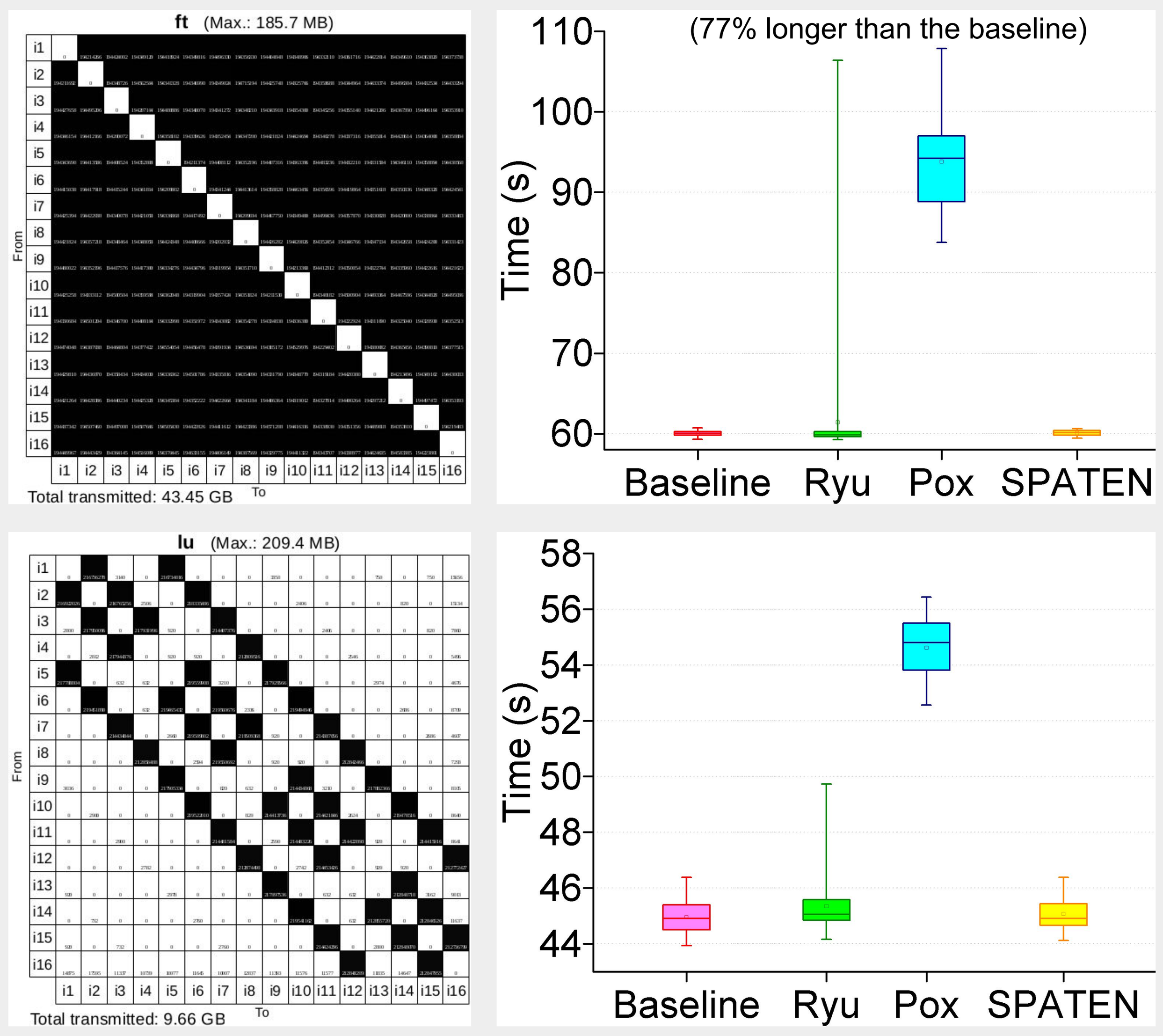
Real testbed composed of Pica8 P-3290 switches

NAS Parallel Benchmarks (http://www.nas.nasa.gov/publications/npb.html)

Impact of Network Programmability

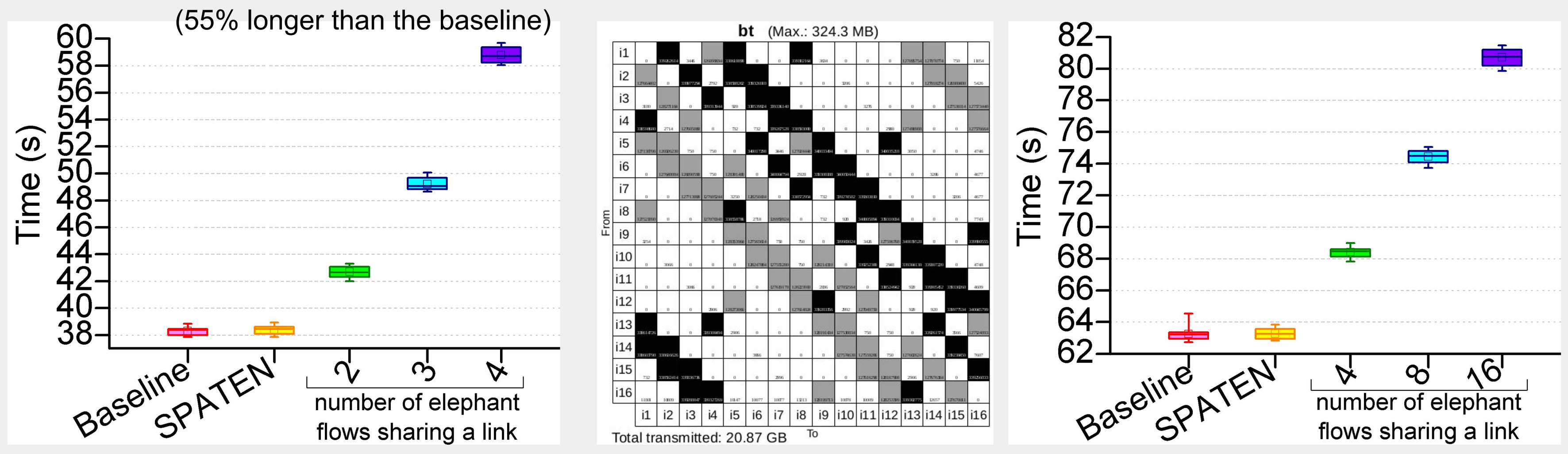
Learning Switch App

- Single switch programmed with SPATEN, Ryu, and Pox
- Most rule-intensive applications *ft* and *lu*



Balancing the Elephant Flows

- Baseline: single switch topology
- Applications that most exchanged data *cg* and *bt*



Conclusion

SPATEN shows that the spatial communication patterns can be used for optimizing the SciApps by identifying the elephant flows and proactively allocating them in a balanced way. SPATEN eliminates the time for querying the controller and reduce the number of installed matching rules, keeping the execution time of SciApps to near-optimal times in a real testbed.

References

[1] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams et al., "The landscape of parallel computing research: A view from berkeley," Tech Rep UCB/EECS-2006- 183, EECS Department, University of California, Berkeley, Tech. Rep., 2006.
[2] M. Afaq, S. Rehman, and W.-C. Song, "Large flows detection, marking, and mitigation based on sflow standard in sdn," Journal of Korea Multimedia Society Vol, vol. 18, no. 2, pp. 189–198, 2015.
[3] R. Trestian, G. M. Muntean, and K. Katrinis, "Micetrap: Scalable traffic engineering of datacenter mice flows using openflow," in 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), May 2013, pp. 904–907.
[4] J. Ru, S. Wei, and Z. Hongke, "Traffic matrix-based routing optimization," in Proceedings of the 2015 International Conference on Computer Science and Intelligent Communication, 2015, pp. 429–432.

UFPR UNIVERSIDADE FEDERAL DO PARANÁ
UFES UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CAPES
CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico
Ministério das Comunicações BRASIL UM PAÍS DE TODOS GOVERNO FEDERAL