Orchestrating a brighter world

NEC

NMRG 60th Meeting - IETF 110

Automated Performance Evaluation of Intent-based Virtual Network Systems

March 8, 2021

NEC System Platform Research Laboratories

Kazuki Tanabe, Tatsuya Fukuda and Takayuki Kuroda

Orchestrating a brighter world

NEC creates the social values of
safety, security, fairness and efficiency
to promote a more sustainable world
where everyone has the chance to reach
their full potential.

Background & Purpose

Background

Automation in ICT System Integration (SI)

- Requirement-definition, Designing and Deployment Phase: Several studies [1][2]
- Testing(Evaluation) phase
  - App-level: CI/CD tools developed (ex: Jenkins) & studies on Web frameworks [3][4]
  - **NW-level: Few studies**

Intent-based Networking (IBN)[5]

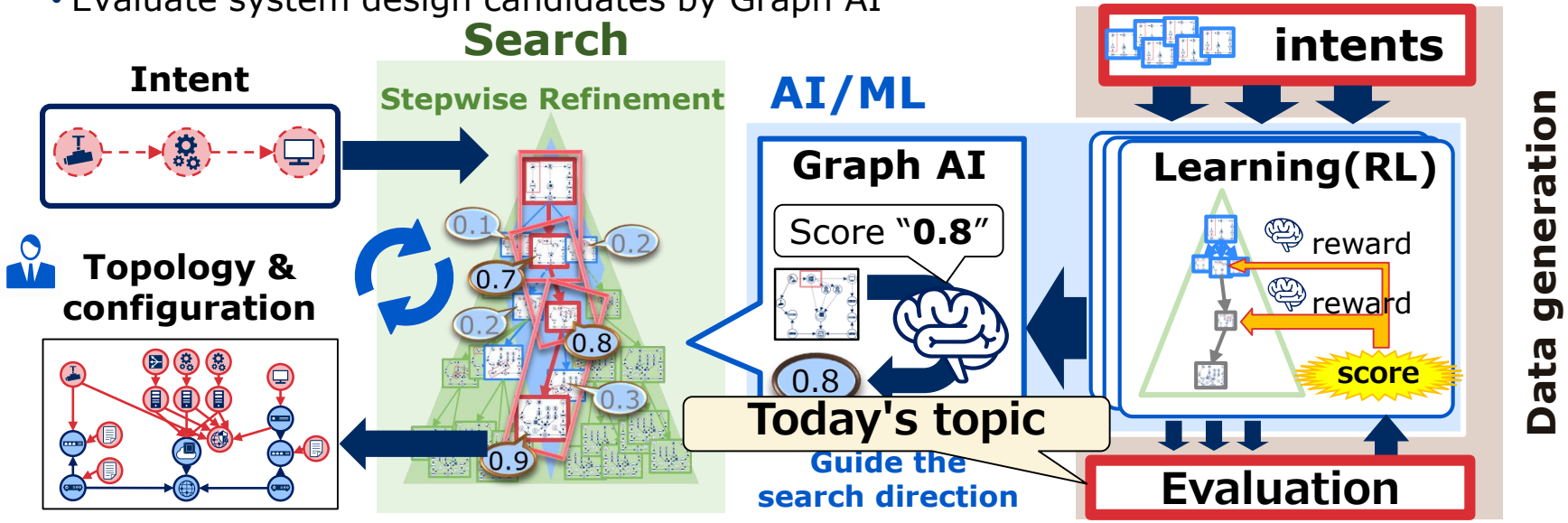
- Network management based on users' abstract intents for system
- Translate intended functions (**what**) to actual NW configurations (**how**)

[1] T. Kuroda et al., "Weaver: A Novel Configuration Designer for IT/NW Services in Heterogeneous Environments," in Proc. IEEE GLOBECOM 2019, Dec. 2019, pp. 1-6.
[2] T. Kuroda, M. Nakanoya, A. Kitano and A. Gokhale, "The Configuration-Oriented Planning for Fully Declarative IT System Provisioning Automation," in Proc. IEEE/IFIP NOMS 2016, Apr. 2016, pp. 808-811.
[3] F. Wang and W. Du, "A Test Automation Framework Based on WEB," in Proc. IEEE/ACIS ICCIS 2012, June 2012, pp. 683-687.
[4] Xiao-yang Guo, Ying-hui Chen, Xue-song Qiu and Fan Tang, "Design and implementation of performance testing model for Web Services," in Proc. CAR 2010, Mar. 2010, pp. 353-356.
[5] Cisco white paper, "Intent-based Networking", https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/digital-network-architecture/nb-09-intent-networking-wp-cte-en.pdf

Our architecture

Weaver[1]: AI-empowered network system designer

- Flexible, rapid & efficient NW provisioning with AI/ML
  - Refine abstract parts in intent
  - Evaluate system design candidates by Graph AI



[1] T. Kuroda et al., "Weaver: A Novel Configuration Designer for IT/NW Services in Heterogeneous Environments," Proc. IEEE GLOBECOM 2019, Dec. 2019, pp.1-6.

Design & Deployment by Weaver

Define concrete system design from abstract requirements (intents)

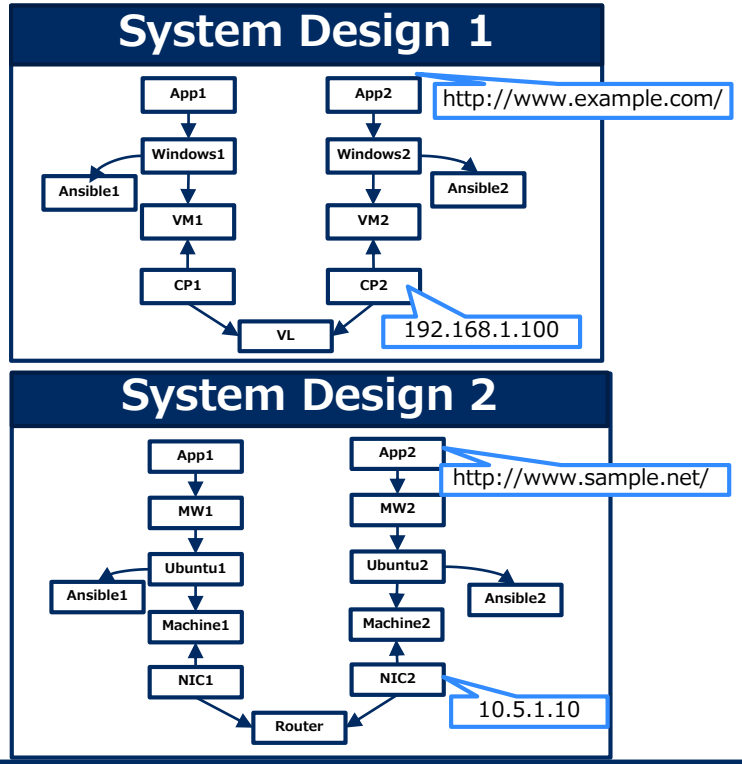
- Multiple configurations can be designed
  - Virtual / Physical
  - Native App / Middleware-based App
  - etc.

System Requirements (Intents)

HTTP[App->App]
Bandwidth[App-App]≧500Mbps

App1 - - - - - - - - - - App2

Design

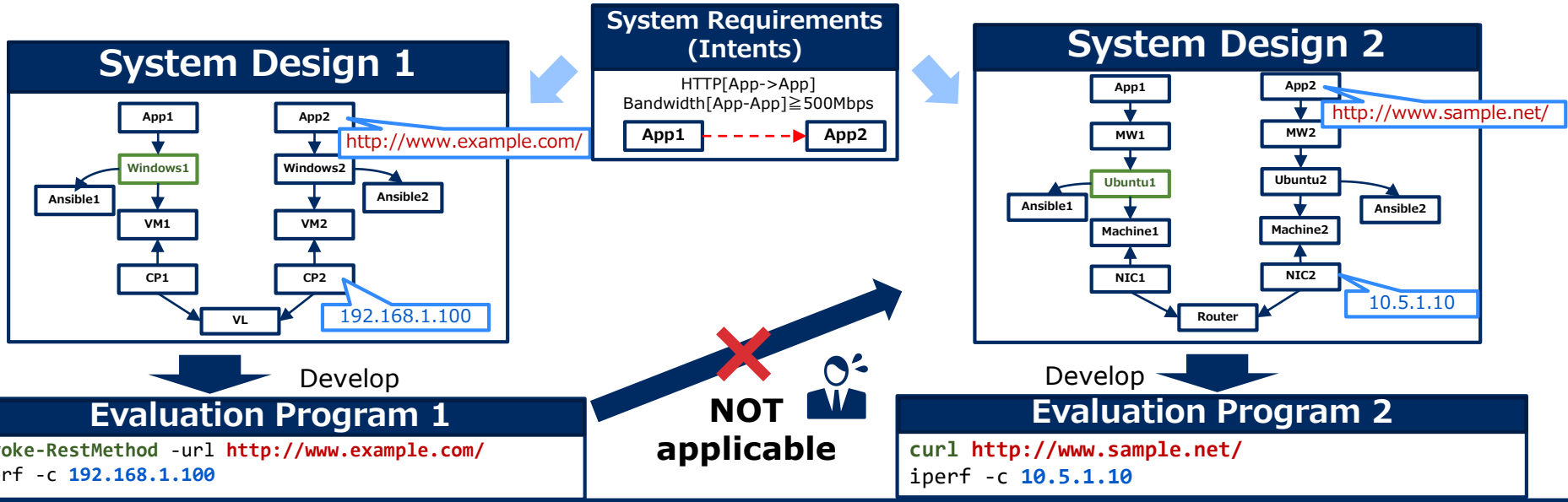


Testing Phase in IBN-Based SI

Development of Evaluation Programs

- Depends on each system design
  - Packages & Commands for each OS
  - Syntax parameter (ex: IP address, URL)

→ **Heavy Burden**

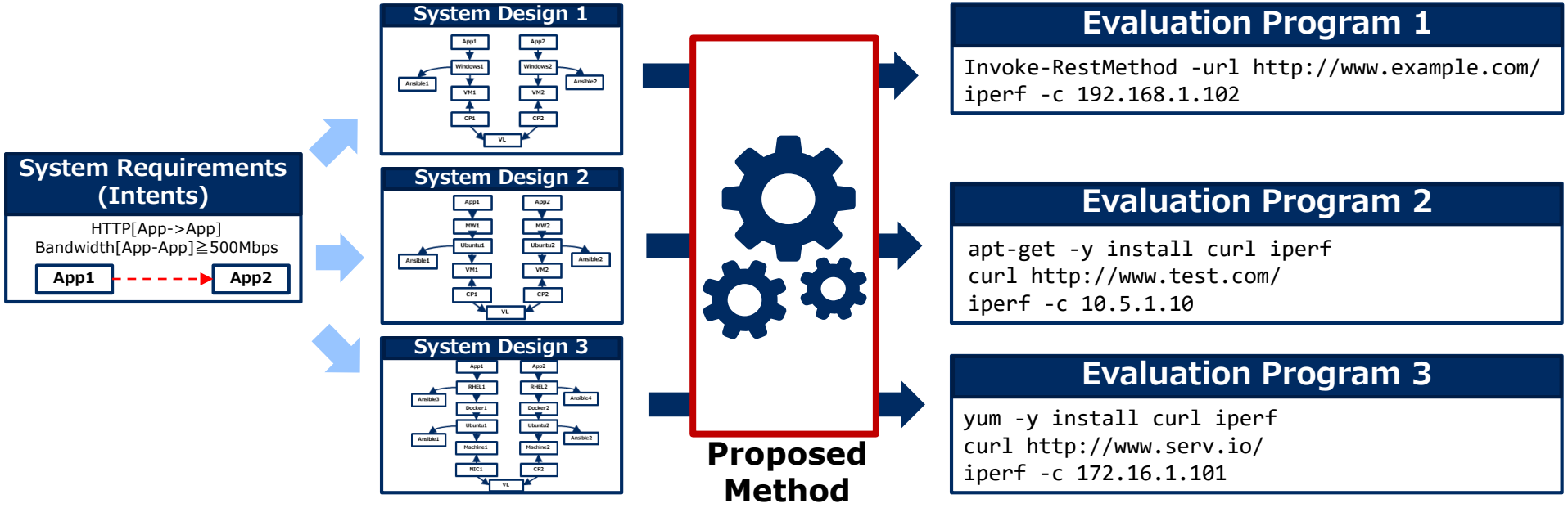


Purpose of Study

Automate Testing Phase of IBN-based SI

Proposal of Evaluation program auto-generation method

- Flexible program generation from abstract intents
- Reduce engineers' burden & accelerate IBN-based SI

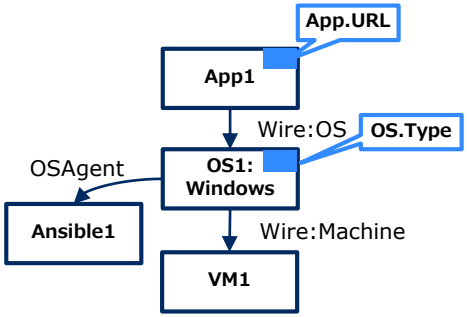


Proposed Method

Model Settings

System Design G = (V, E)

- Directed graph composed of nodes V and edges E
  - Node v ∈ V : system components (OS, App, Router etc.)
    - contains properties
  - Edge e ∈ E : directed connection between nodes

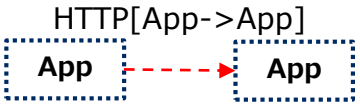


System Requirements $G^{req} = (V^{req}, E^{req})$

- Directed graph composed of nodes V^{req} and edges E^{req}
- A part of system design
- Abstract expression of users' requirement for systems

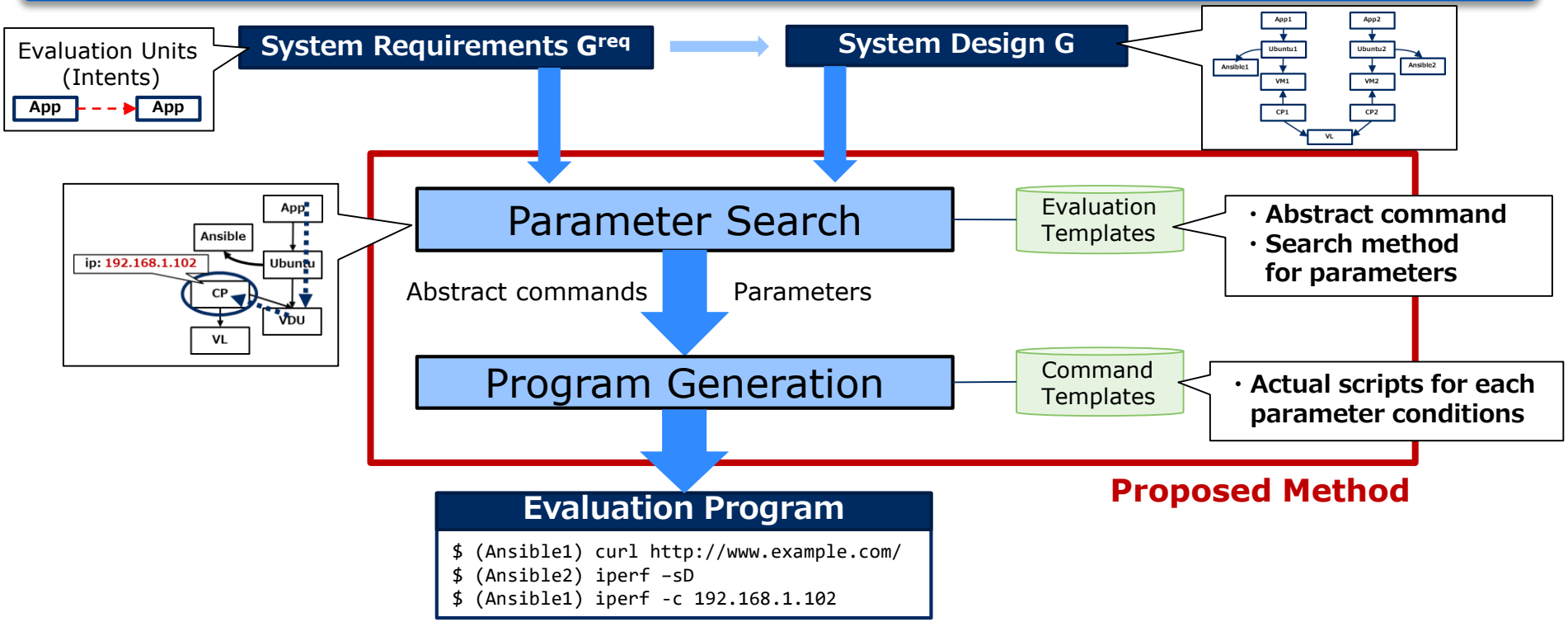
Evaluation Units (Intents)

- Express as abstract nodes or edges



Outline of Proposed Method

Concretizes abstract intents into executable commands by referring templates & acquiring parameters on the graph



Parameter Search Step

Evaluation Unit

Bandwidth[App-App]≧500Mbps

System Requirements

HTTP[App->App]
Bandwidth[App-App]≧500Mbps

App1 - - - - - - - App2

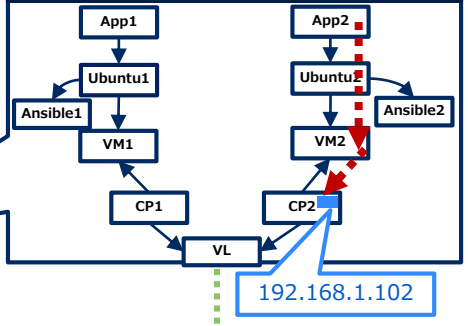
Evaluation Templates

Abstract command

bandwidth(Agent1, OS1.Type, Agent2, OS2.Type, CP.ip, x)

apply
search method

System Design



Abstract command with parameters

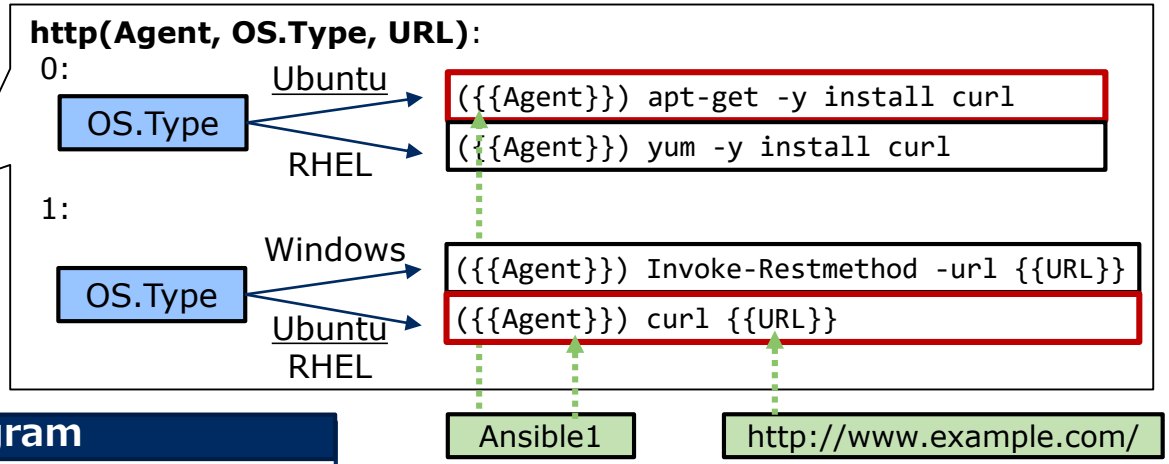
bandwidth(**Ansible1**, **Ubuntu**, **Ansible2**, **Ubuntu**, **192.168.1.102**, **500**)

Program Generation Step

Abstract command with parameters

http(Ansible1, Ubuntu, http://www.example.com/)
bandwidth(Ansible1, Ubuntu, Ansible2, Ubuntu, 192.168.1.102, 500)

Command Templates

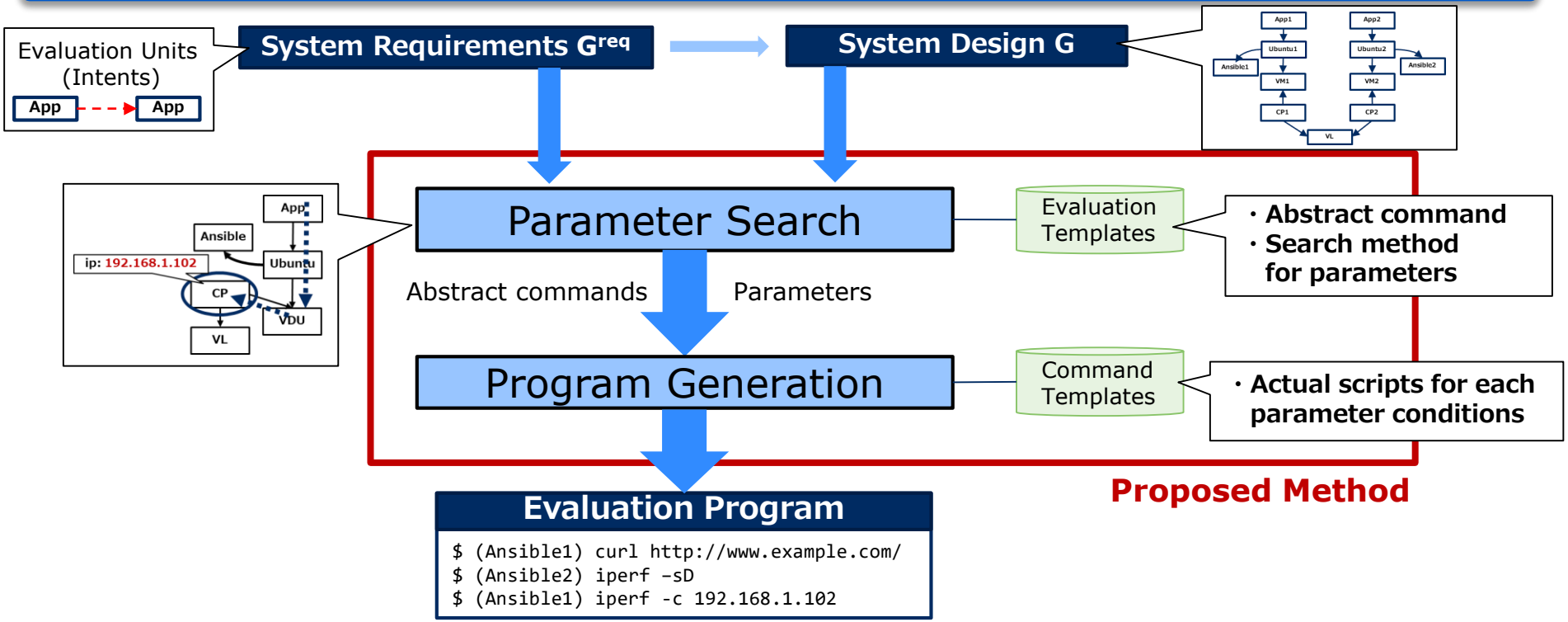


Evaluation Program

```
# HTTP[App->App]
$ (Ansible1) apt-get -y install curl
$ (Ansible1) curl http://www.example.com/
# Bandwidth[App-App]≧500Mbps
$ (Ansible1) apt-get -y install iperf
$ (Ansible2) apt-get -y install iperf
$ (Ansible2) iperf -sD
$ (Ansible1, bandwidth>=500Mbps) iperf -c 192.168.1.102
```

Outline of Proposed Method (repeated)

Concretizes abstract intents into executable commands by referring templates & acquiring parameters on the graph



Evaluation

Evaluation

Verify Evaluation Programs Generated by Proposed Method

- Evaluation programs are generated as workflows of test tasks
- Systems deployed on Openstack virtual environment
- Ansible agents execute evaluation scripts

Evaluation Environment 1

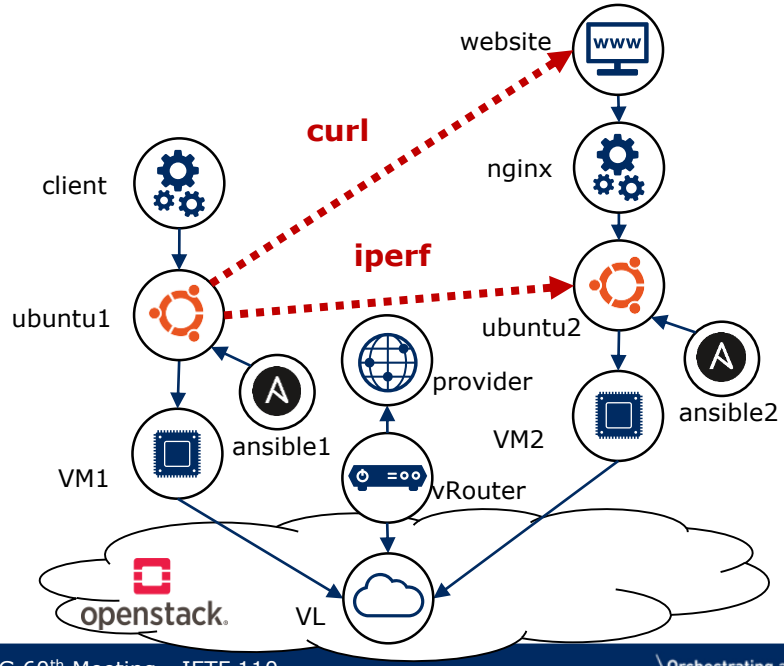
Deployed a system design on Openstack virtual environment

- Executed evaluation scripts for 2 evaluation units
  - HTTP[App->App]: An HTTP request by curl
  - Bandwidth[App->App]≧500Mbps: Bandwidth measurement by iperf

System Requirements

HTTP[App->App]
Bandwidth[App-App]≧500Mbps

App1 - - - - -> App2



Results on Evaluation Environment 1

Both evaluation units determined as **PASS**

- HTTP[App->App] (curl): regular HTTP response
- Bandwidth[App->App]≧500Mbps (iperf): Measured bandwidth 818Mbps

Evaluation Units	Result	Program Output

HTTP[App->App]	**PASS**	`<!DOCTYPE HTML>`
		`<html>`
		`<head>`
		`<title>Welcome to nginx!</title>`
		`...`

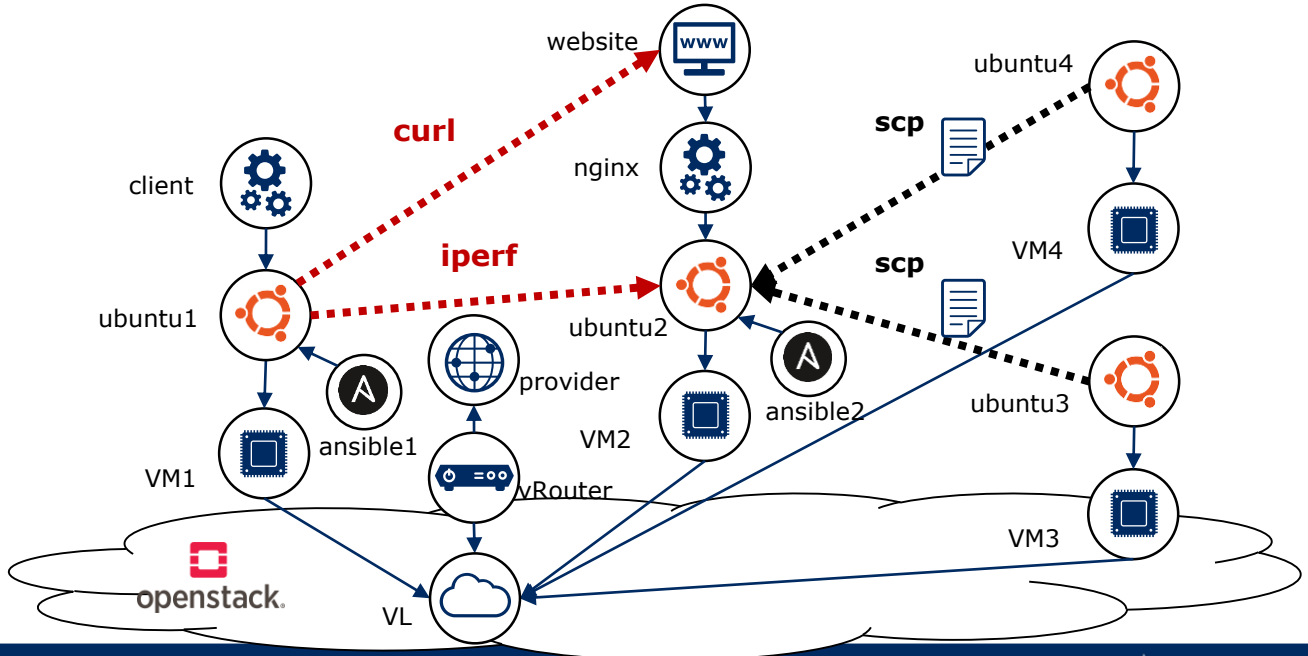
Bandwidth[App-App]≧500Mbps	**PASS**	`[ 3] local 192.168.1.101 port 45407 connected with 192.168.1.102 port 5001`
		`[ ID] Interval Transfer Bandwidth`
		`[ 3] 0.0-10.0 sec 977 MBytes 818 Mbits/sec`

Satisfaction of system requirements
can be **automatically** determined!

Evaluation Environment 2

Addition of 2 VMs

- Create background traffic
  - scp file transfers: ubuntu3 -> ubuntu2 & ubuntu4 -> ubuntu2
- Execute the same evaluation program as Evaluation Environment 1



Results on Evaluation Environment 2

FAIL for the bandwidth constraint

- Bandwidth[App->App]≧500Mbps (iperf): Measured bandwidth 306Mbps
  - Throughput reduction by background traffic
  - Performance reduction of ubuntu2 node by processing file transfers

Evaluation Units	Result	Program Output
HTTP[App->App]	**PASS**	<!DOCTYPE HTML> <html> <head> <title>Welcome to nginx!</title> ...
Bandwidth[App->App]≧500Mbps	**FAIL**	[3] local 192.168.1.101 port 45812 connected with 192.168.1.102 port 5001 [ID] Interval Transfer Bandwidth [3] 0.0-10.0 sec 366 MBytes 306 Mbits/sec

Performance can be determined **per evaluation unit**
Easy detection for components to be improved

Conclusion

Conclusion

Summary

- Graph search-based evaluation program generation method
  - Modeled system requirements & designs as graph structures
  - Parameter acquisition in a system design graph by search methods

Future Works

- Expansion for more complex topology
  - ex: Machines with multiple NIC and IP addresses


Orchestrating a brighter world
NEC