Spatial model checking with microservices for the internet-of-things

March 15, 2018

Politecnico di Milano, Italy

Setting

- Wide deployments of internet-enabled things within smart applications (i.e. smart cities)
- Complex relations between locations of things with respect to their environment may arise
- Things change location; location may affect satisfaction of system requirements

Setting

- Wide deployments of internet-enabled things within smart applications (i.e. smart cities)
- Complex relations between locations of things with respect to their environment may arise
- Things change location; location may affect satisfaction of system requirements

How can we enable runtime verification of requirements of space-dependent systems of internet-of-things?

Setting

- Wide deployments of internet-enabled things within smart applications (i.e. smart cities)
- Complex relations between locations of things with respect to their environment may arise
- Things change location; location may affect satisfaction of system requirements

How can we enable runtime verification of requirements of space-dependent systems of internet-of-things?

- Properties expressing arbitrary relations in space are known upfront
- We seek formal assurances that properties hold in a given space-things configuration

Verify spatial properties on configurations of city space populated with locations of things at given point

Key intuition: Space is composed of relations between entities

- Evaluation model is a discrete graph structure representing a topological view of space
- Serves as the critical abstraction step
- Enables formal specification and verification techniques

Expressing & Verifying Properties

A spatial logic can express complex properties of space and enable automated verification. For example:

Example

From the location where a taxi is near a gas station, is there a way to go to a metro station going through –an arbitrary number of– bus stops?

Example

From which locations where taxis are near a pharmacy, we can reach a bar through metro stations and bus stops, without going near a hospital?

We can support logic-based reasoning

- Formal specification of properties
- Spatial properties may be non trivial

Spatial Evaluation Model

1. Generate a closure space from the accessibility graph
2. Equip with a valuation function associating points with atomic propositions (POIs, taxis etc)
3. Big-step semantics: truth values of spatial relations or points where spatial relations hold

Spatial Logic for Closure Spaces[1]

$$\tau ::= \mathsf{p} \mid \top \mid \neg\tau \mid \tau \wedge \tau \mid \mathcal{C}\ \tau \mid \tau\ \mathcal{S}\ \tau$$

- Closure "one step" modality
- Surrounds modality

[1] Vincenzo Ciancia et al. "Specifying and verifying properties of space". In: *Theoretical Computer Science*. Springer, 2014, pp. 222–235.

Modelling complex properties in space[2]: reachability

Notion of path in a closure space

$$\phi \, \mathcal{R} \, \psi \stackrel{def}{=} \neg\big((\neg\psi) \, \mathcal{S} \, (\neg\phi)\big)$$

Constrain the traversed path

$$\phi \, \mathcal{T} \, \psi \stackrel{def}{=} \phi \wedge \big((\phi \vee \psi) \, \mathcal{R} \, \psi\big)$$

Predicate on entities encountered on the traversed path

$$\phi \, \Re(\psi) \, \zeta \stackrel{def}{=} \phi \, \mathcal{T} \, \big((\psi \, \mathcal{T} \, \zeta) \wedge (\psi \, \mathcal{T} \, \phi)\big)$$

Nearness as nested applications of closure operator

$$\mathcal{N}_n \phi \stackrel{def}{=} C_n C_{n-1} .. \phi$$

[2] Christos Tsigkanos, Timo Kehrer, and Carlo Ghezzi. "Modeling and Verification of Evolving Cyber-Physical Spaces". In: *ACM SIGSOFT Symposium on the Foundations of Software Engineering*. 2017, pp. 38–48.

Example

From the location where a taxi is near a gas station, is there a way to go to a metro station going through –an arbitrary number of– bus stops?

$$\text{taxi } \Re(\mathit{transportbusstop}) \text{ transportsubway} \wedge \mathcal{N}_n \text{transportfuel}.$$

Example

From which locations where taxis are near a pharmacy, we can reach a bar through metro stations and bus stops, without going near a hospital?

$$\text{taxi } \Re(\mathit{transportbusstop} \vee \mathit{transportsubway} \wedge !(\mathcal{N}_n \mathit{healthhospital}))$$
$$\text{foodbar} \wedge \mathcal{N}_n \text{healthpharmacy}.$$

Intuition:

- Trajectory datasets can provide movement data points
- Points-of-interest datasets can provide static graphs

Combining both

Movement data points over time, over a closure space

Jing Yuan et al Dataset

T-Drive trajectory dataset[3]

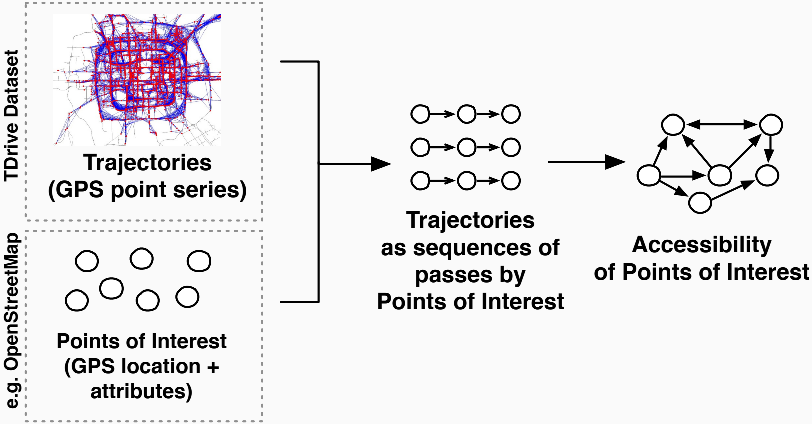
- trajectories of 10k taxis over one week
- 15M data points
- 9M kilometers total distance

OpenStreetMap points-of-interest in Beijing

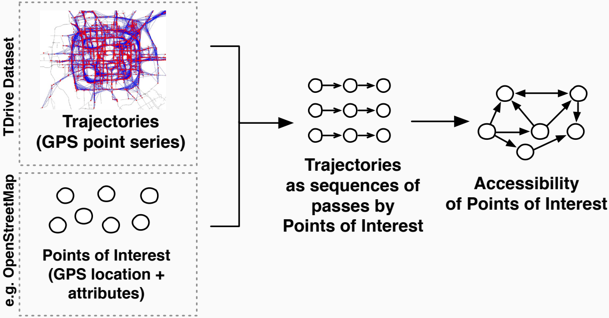
- Various landmarks, public services, shops, etc
- e.g. 11741 POIs

[3] Jing Yuan et al. “Driving with knowledge from the physical world”. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 316–324, Jing Yuan et al. “T-drive: driving directions based on taxi trajectories”. In: *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*. ACM. 2010, pp. 99–108.

Obtaining a topological model: process



Obtaining a topological model: process



1. T-Drive trajectory dataset
2. Points-of-Interest
3. Discover points that trajectories pass through

Key idea: if an entity's trajectory passes from a point to another, then these points are connected

- We obtain connectivity of points as a graph structure
- Nodes are Points-of-Interest
- Edges reflect "accessibility" of a node from another

Evaluation Dataset

Discrete presences of taxis in Beijing over time

- Taxis move between various points of interest
- Can consider them as position *updates* to the global topological space of Beijing

Intuition

Every taxi presence triggers a check of a global spatial property

Evaluation Dataset: A property

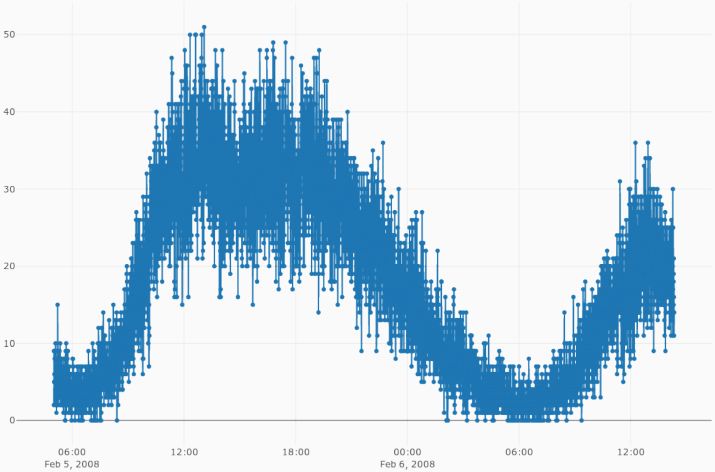
"Always from where taxis are near metro stations, there is a way to go to gas stations going through –an arbitrary number of– bus stops or bars "

Formulation

$(\mathrm{taxi} \wedge \mathcal{C}\mathrm{transportsubway}) \wedge \neg((\neg(((\mathrm{transportbusstop} \vee \mathrm{foodbar}) \wedge \neg((\neg(\mathrm{transportfuel}))\mathcal{S}(\neg((\mathrm{transportbusstop} \vee \mathrm{foodbar}) \vee (\mathrm{transportfuel})))) \wedge ((\mathrm{transportbusstop} \vee \mathrm{foodbar}) \wedge \neg((\neg(\mathrm{taxi} \wedge \mathcal{C}\mathrm{transportsubway}))\mathcal{S}(\neg((\mathrm{transportbusstop} \vee \mathrm{foodbar}) \vee (\mathrm{taxi} \wedge \mathcal{C}\mathrm{transportsubway}))))))\mathcal{S}(\neg((\mathrm{taxi} \wedge \mathcal{C}\mathrm{transportsubway}) \vee (((\mathrm{transportbusstop} \vee \mathrm{foodbar}) \wedge \neg((\neg(\mathrm{transportfuel}))\mathcal{S}(\neg((\mathrm{transportbusstop} \vee \mathrm{foodbar}) \vee (\mathrm{transportfuel})))) \wedge ((\mathrm{transportbusstop} \vee \mathrm{foodbar}) \wedge \neg((\neg(\mathrm{taxi} \wedge \mathcal{C}\mathrm{transportsubway}))\mathcal{S}(\neg((\mathrm{transportbusstop} \vee \mathrm{foodbar}) \vee (\mathrm{taxi} \wedge \mathcal{C}\mathrm{transportsubway}))))))))$

(i.e. not trivial -;)

Beijing Composite Dataset: Lunchtime Requests Workload



Experimental Setup

We deployed a model checker (computation logic) and the current topology (static state) as a RESTful microservice.
Three alternative architectures:

- **Lambda Functions**: 3Gb RAM (vCPU quotas are assigned accordingly)
- **Single Instance of a Big VM**: Standard F64s_v2 (64 vcpus, 128 GB memory)
- **Autoscaling Group**: 5 to 20 t1.micro VMs (1Gb Ram, 1vCPU each)

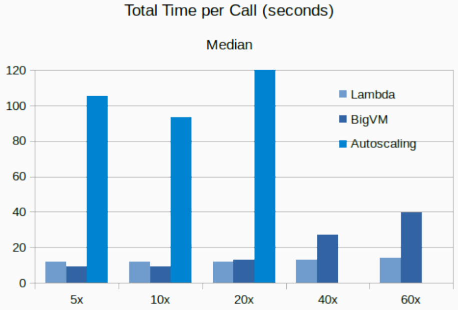
Positions tracker (dynamic state) deployed as another standalone microservice

Execution Sample

- **One hour window** of the Beijing dataset as workload (536 data points), the beginning of a daily peak
- **Time multipliers**: To simulate events that happen closer to each other.
  - 5x / 10x / 20x / 40x / 60x time multipliers.
- Run the experiment combining the different time multipliers with the three alternative deployments

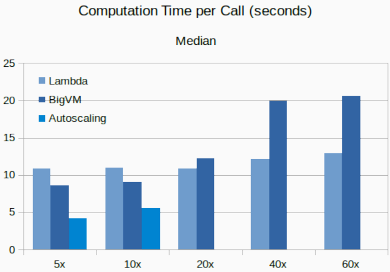
Service Level Agreement

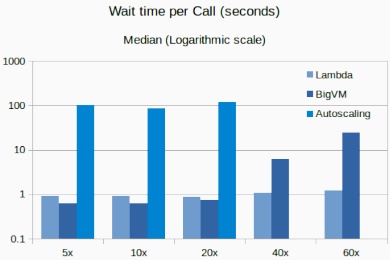
Defined a SLA of 30 seconds/call maximum for responding to model checking requests

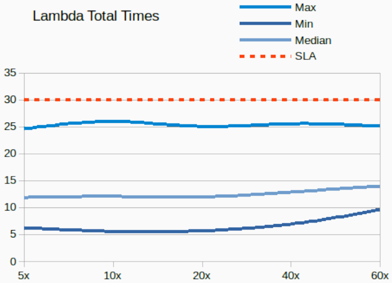


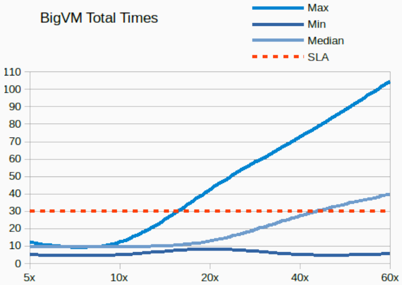
- The Autoscaling group had the **worst performance**. Requests timed out (504) with 20x time or more.
- Lambda held an almost **constant performance** (12s/call) for all workloads
- The BigVM **(baseline)** performed better (9-12s) up to 20x, then started to decrease performance.

Experimental Results






Lambda Total Times
Max
Min
Median
SLA
35
30
25
20
15
10
5
0
5x
10x
20x
40x
60x


BigVM Total Times
Max
Min
Median
SLA
110
100
90
80
70
60
50
40
30
20
10
0
5x
10x
20x
40x
60x

Experimental Results: Discussion

- Lambda functions kept a **constant performance** even under dense workloads, being more reactive and scalable.
- The BigVM used as a baseline performed well for medium workloads, but not against peaks or more dense workloads **(e.g. workloads unknown beforehand)**.
- The autoscaling group did not seem suitable, but more experimentation is needed (fine-tuning parameters, scaling containers?).
- Cost analysis to come (if the workload is more or less known, lambda could become more costly in comparison)