## Introduction

The main focus of graph analytics conventionally has been on static graphs which encompass the idea of capturing the story of each data set in disconnected windows of time. While this application can be useful, it ignores the context of how the transformation for each instance of the graph came to be. Raphtory aims to provide a way to analyse data efficiently, offering dynamic graphs that sync automatically with streamed or stored data. Raphtory gives the option to dive deep into the analysis and explore the changes in structural (vertex and edge properties) and temporal (time of when components are created, updated, and deleted) scopes of the graphs. This opens up many possibilities including tracking cryptocurrency fraud, monitoring activities on social media, or COVID transmission patterns.

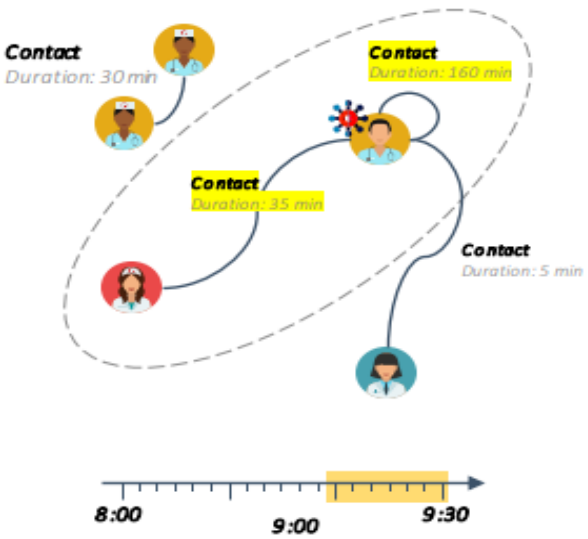
### What are temporal graphs?

Temporal graphs are graphs that change with time. Specifically, the relationships between the entities (vertices and edges) in a temporal graph change but the entities themselves do not. In this sense, temporal graphs are presented like a series of static graphs so that each update creates a new static graph for viewing. Temporal graphs are like movies in many ways: the landscape of the story (the graph) transforms as new information and data come in. Below is a use case example:

### Tracking the spread of COVID with Raphtory

A simple example on how temporal graph analytics can help in real-world instances, in this case, preventing COVID outbreak at a hospital.

Medical staff with ID 1234 has COVID and the hospital needs to act quickly to contact trace those who have been exposed to the virus. Since the virus can be transmitted through contact of more than 30 minutes, the hospital can use temporal graphs to see the network of exposure across time to evaluate the impact given that staff members are equipped with location sensors.



## Getting Started

Let's use an example Raphtory project and install SBT (Scala Build Tool) to get it up and running.

1. Clone the example and get the latest release.
2. Download the `raphtory.jar` here. Move it into the `lib` directory of the example project.
3. Install SBT by following their guide. The example project uses SBT to compile the source code.
4. Initiate SBT by changing into the example project directory in the terminal and running the command `sbt`. The SBT interactive shell starts once it shows `>`.
5. Run `compile` to build the project.
6. Run `run` to analyse the project. Let's explore the output of the analysis on the next page.

## Cool, it works! But what does the output mean?

Here are a few main points as seen below with the output from the example::

- Raphtory has started:

```
Cluster Node [akka.tcp://Citation-system@192.168.2.2:1600] - Started up successfully
```

```
RaphtoryReplicator - Router 0 has come online.
RaphtoryReplicator - Partition Manager 0 has come online.
```

```
Number of routers: 1
Number of partitions: 1
Cluster ready for Analysis
All data sent
```

- Analysis for the example is running, and this means that Raphtory is working properly on your machine. This is part of the output from the example functions automatically submitted. The output for this is long so here is a snippet of what it should show:

```
{"time":10000,"vertices":84,"edges":262,"degree":3.119047619047619}
View Analysis manager for com.raphtory.algorithms.DegreeBasic_1607467033424 at 10000 finished
```

Go to the documentation for tutorials and explanations on graph building, analysis, and more.

## FAQ

**Which IDE should I use?** Since Raphtory is built with Scala, it is best to use IntelliJ IDE for the code. They have a community version which is free. Follow their guide for installation.

**What should I do if I get error messages indicating failure to ingest data?** Here are a few things to check if you are getting error messages:

- Check if your files are in the right directory. The latest release of Raphtory, `raphtory.jar`, should be inside the `lib` directory.
- The data source is too big or too many states are being generated in the analysis. Try testing on a smaller sample of the data set.

## Ready to learn more?

Check out these links for more information:

- Detailed overview on the research
- Introduction to temporal graphs and see some of the exciting software built on top of Raphtory
- Raphtory article on the Alan Turing Institute site
- Another great use case: investigating the Alt-Right network with temporal graphs
- Follow the latest development on the official blog
- Follow the Raphtory Twitter
- Join the Slack group