



ParTraCI: Una implementación de TraCI en paralelo para simulaciones distribuidas en SUMO

Nicolás Arroyo

Andrés Acosta

Prof. Jorge Espinosa

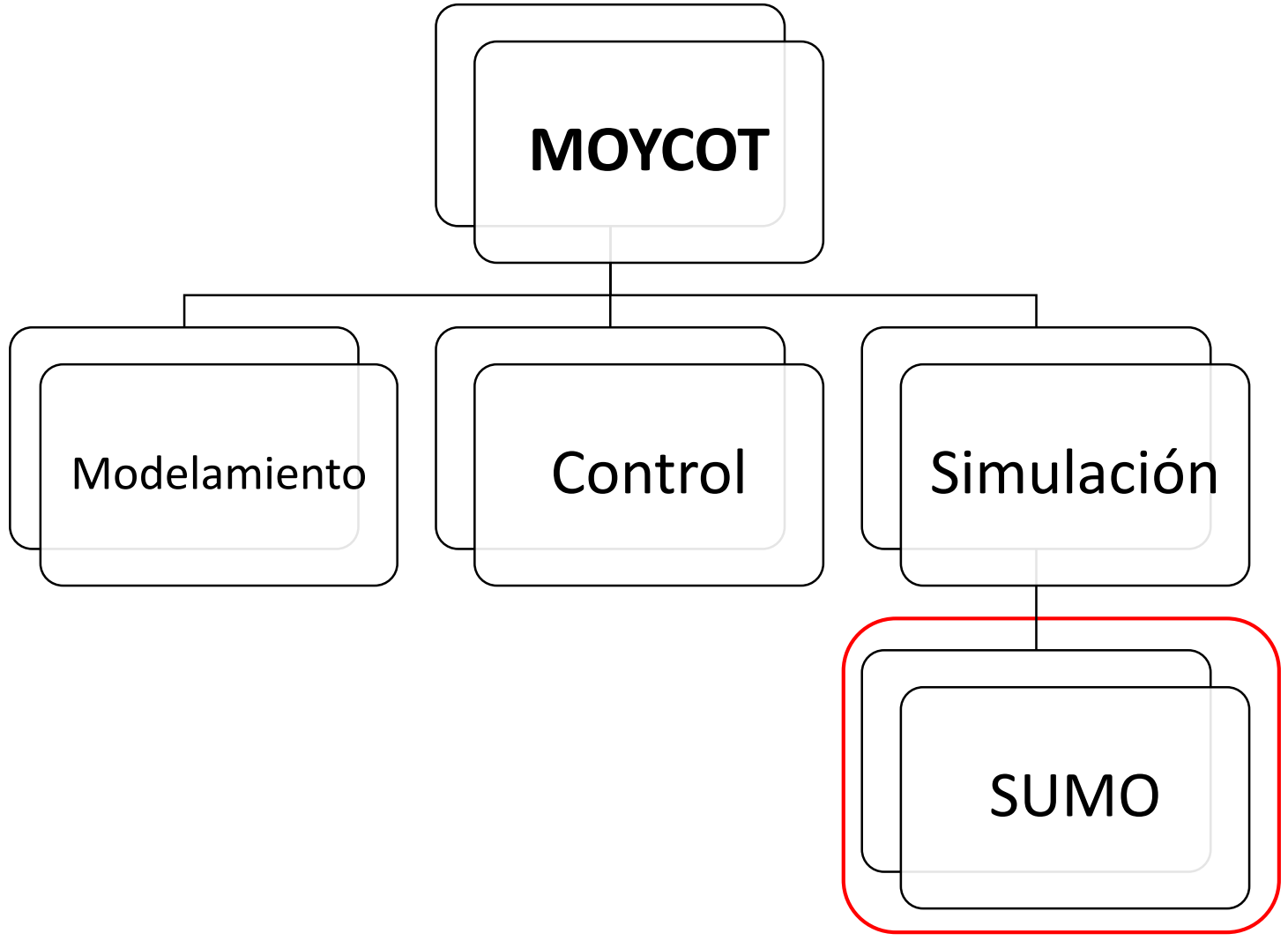
Prof. Jairo Espinosa



MOYCOT

Modelamiento y Control de Tráfico urbano en la ciudad de Medellín

Objective: Desarrollar y validar métodos y herramientas que permitan modelar y controlar el tráfico urbano de una ciudad grande (Medellín), teniendo en cuenta los diferentes actores de tráfico (peatones, vehículos).



Esquema

- ParSUMO
- ParTraCI
- Resultados

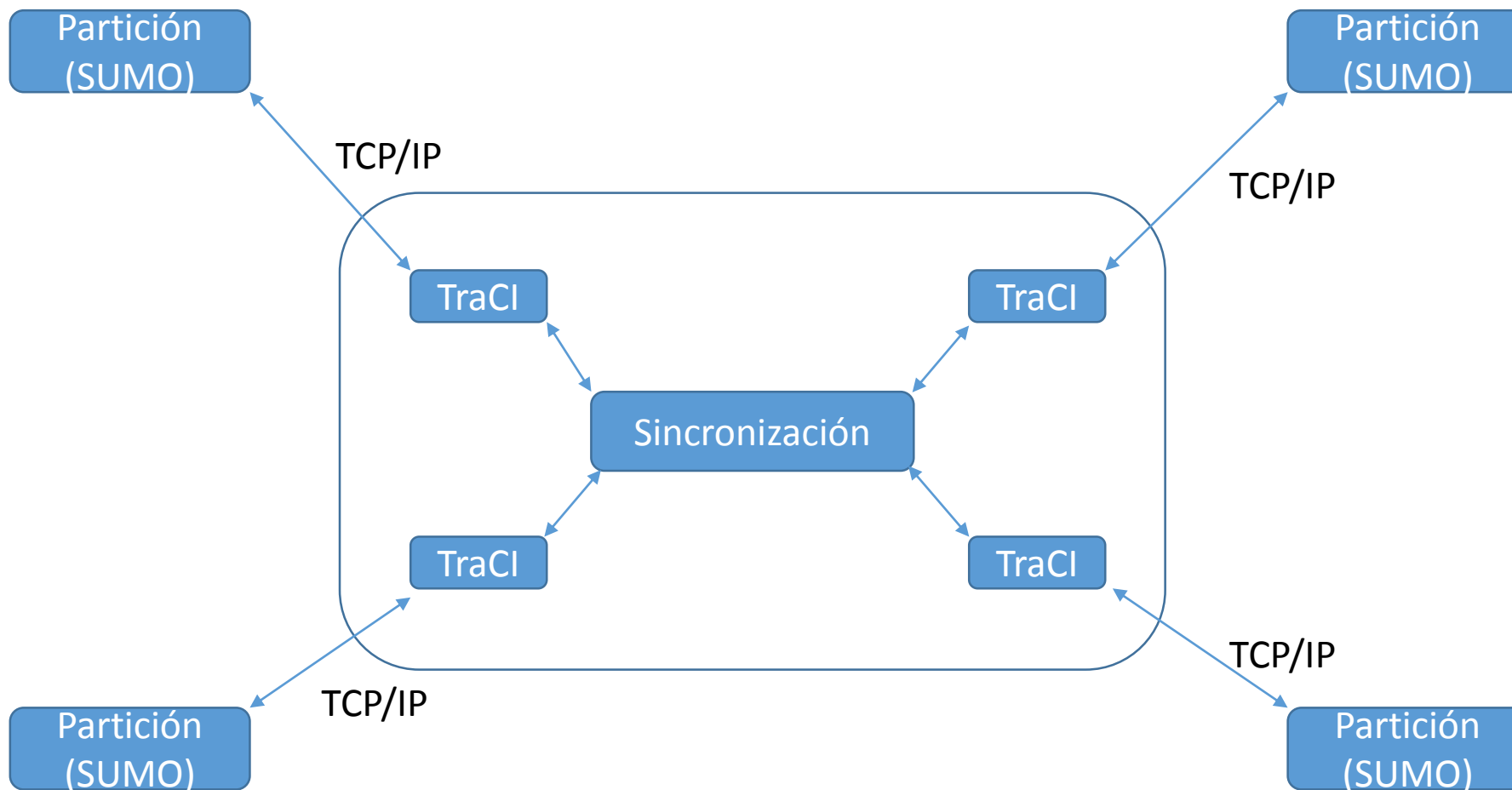
ParSUMO – Simulación distribuida

- Partición
 - Particiones regulares
 - Particiones no regulares
- Comunicación entre particiones (TraCI)
- Sincronización entre particiones (TraCI)

ParSUMO – Simulación distribuida

- **Partición**
 - Escenario con demanda de tráfico igualmente distribuida
 - Particiones rectangulares
 - Cada partición corre como una instancia de SUMO
- **Comunicación**
 - Memoria compartida y TCP-IP con TraCI
 - Una instancia de TraCI por partición corriendo en hilos paralelos

ParSUMO – Simulación distribuida



ParSUMO – Simulación distribuida

- Sincronización
 - Esquema de maestro y esclavo. Los vehículos en la partición esclava están supeditados a los de la partición maestra.
 - Los vehículos en la partición esclava no desaparecen al entrar a la nueva partición, sólo cuando dejan los enlaces comunes.
 - Los vehículos en la partición maestra son ingresados tan pronto entran los enlaces comunes en la partición esclava

ParSUMO – Simulación distribuida

- Sincronización



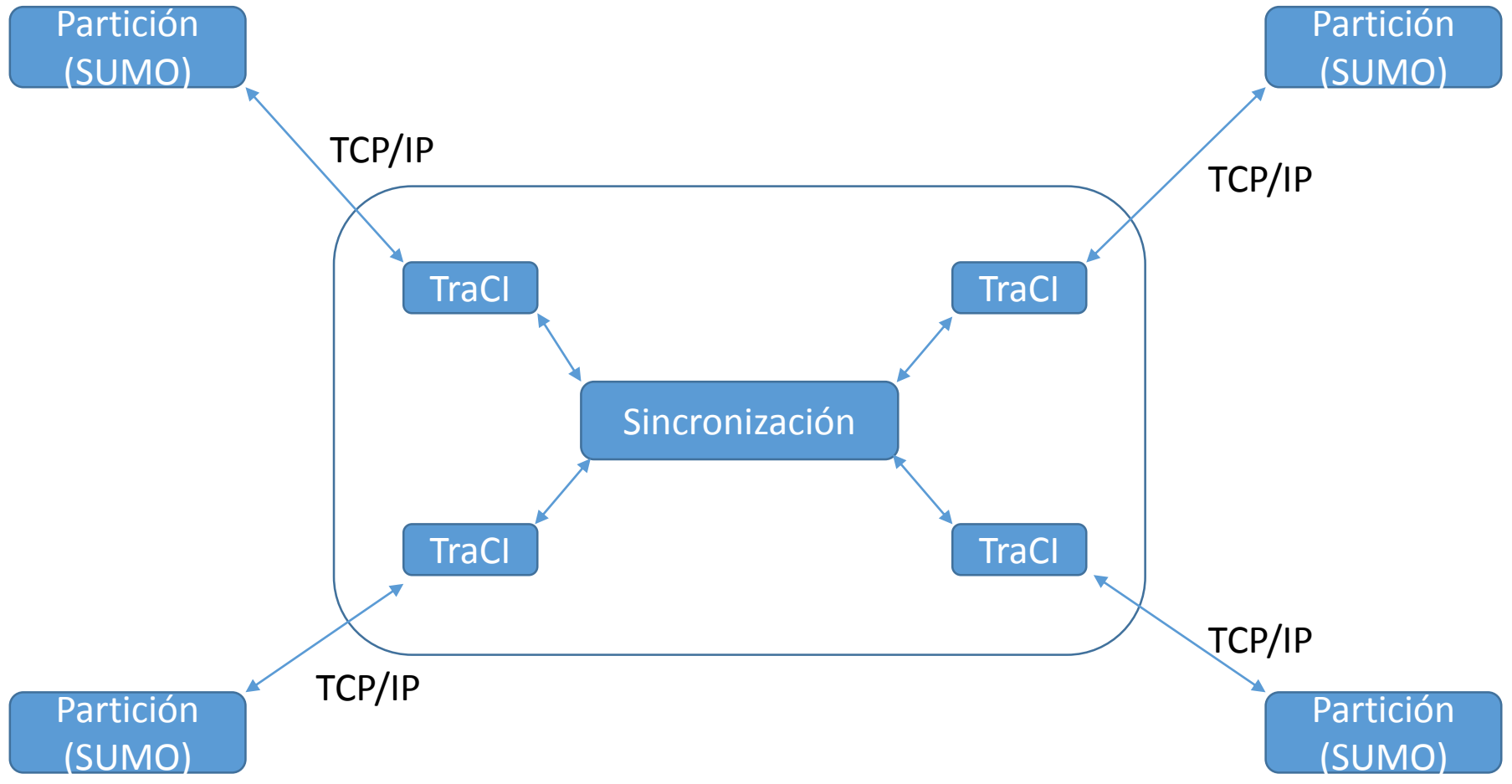
ParSUMO – Simulación distribuida

- Sincronización
 - Se usan los comandos de TraCI **slowDown** y **moveTo** para controlar los vehículos en la partición esclava
 - Todos los enlaces comunes son sincronizados en cada paso de simulación

ParTraCI – Interfaz distribuida

- TraCI con ParSUMO
- ParTraCI
- Funcionalidades

TraCI con ParSUMO



TraCI con ParSUMO

- Se debe manejar una instancia de TraCI por cada escenario
- Cada instancia de TraCI se llama de forma individual

ParTraCI

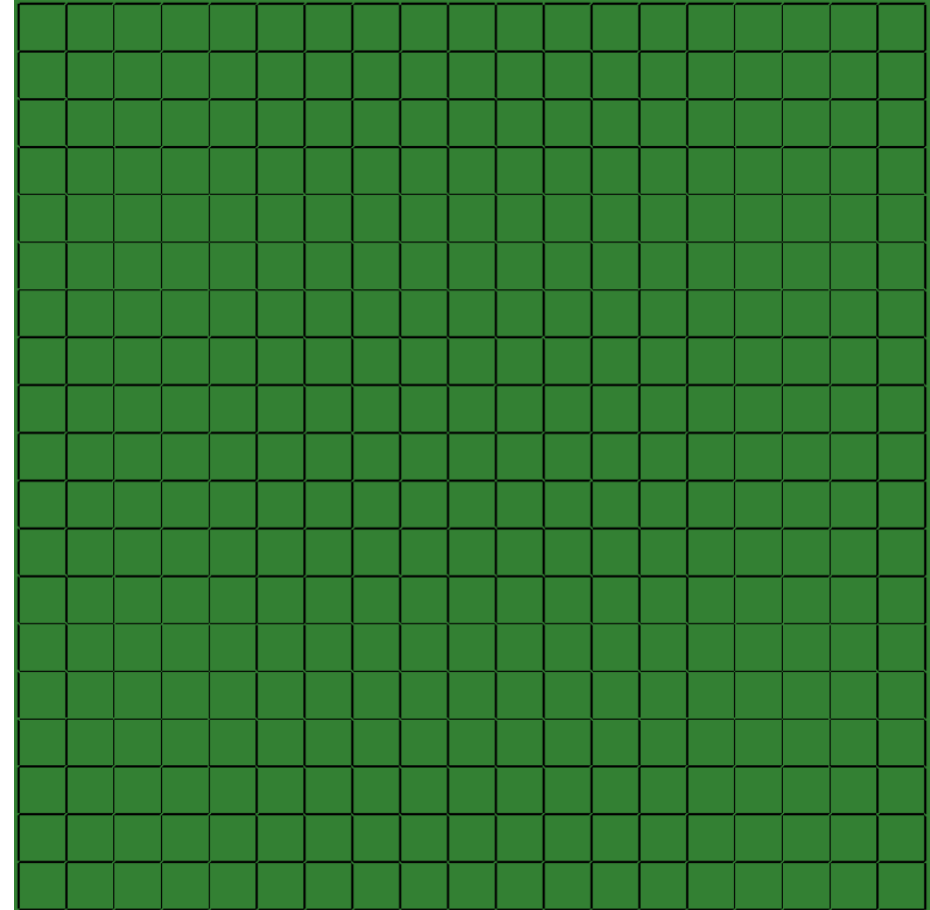
- Las peticiones que se hagan a ParTraCI se hacen a cada una de las instancias de SUMO creadas por ParSUMO
- Cada instancia de TraCI corre en paralelo
- Estructura similar a TraCI
- Desarrollado en C++ con envoltorios (wrappers) de Python, Java y MATLAB

ParTraCI – Funcionalidades

- Obtención de emisiones (monóxido y dióxido de carbono, material particulado, óxidos de nitrógeno, hidrocarburos) y consumo de combustible a nivel de línea, enlace y vehículo
- Control de ciclos de semáforo
- Bucles de inducción

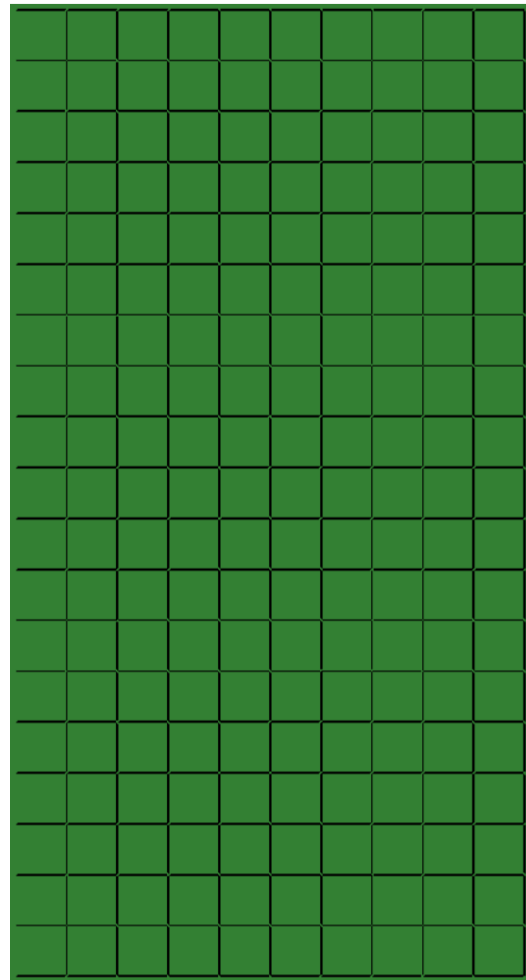
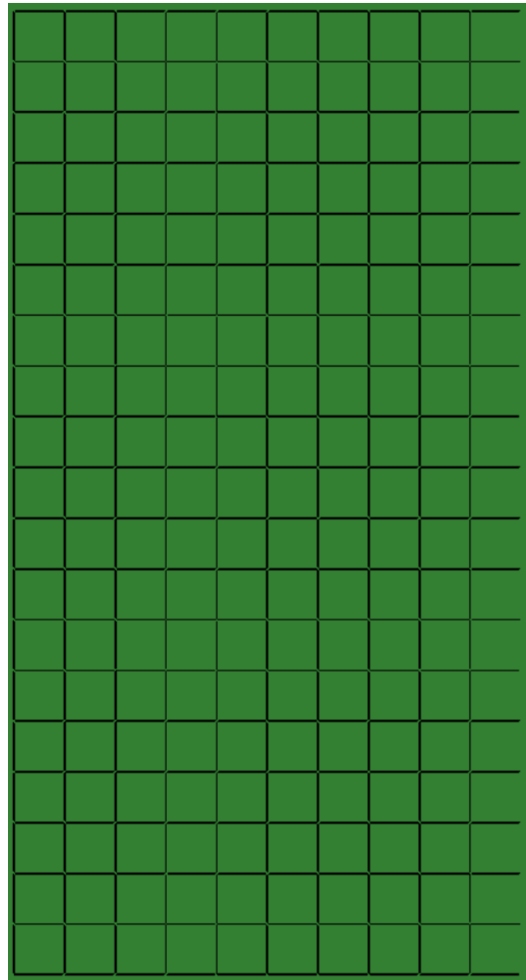
Resultados - escenarios

- Los escenarios se basan en una malla de 20x20
- Se simuló una hora con una demanda promedio de 3300 vehículos por hora
- Cada escenario se simuló 20 veces



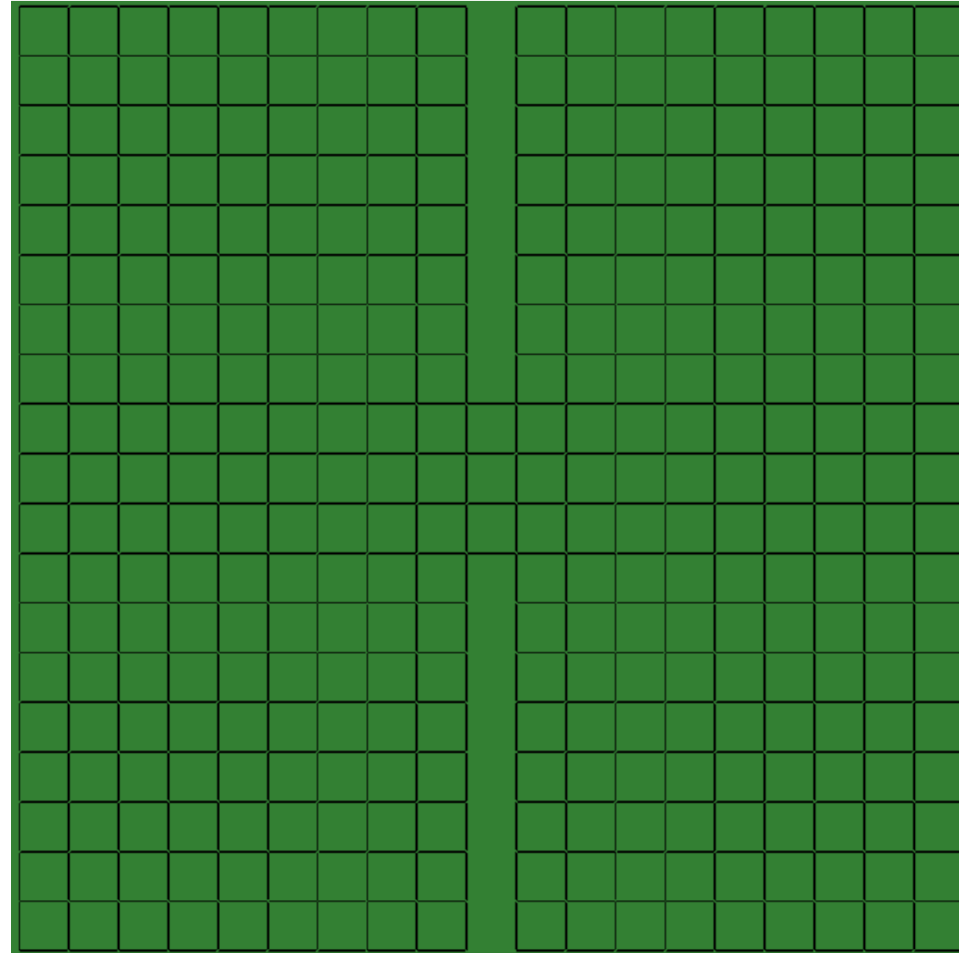
Resultados - escenarios

- Particiones rectangulares



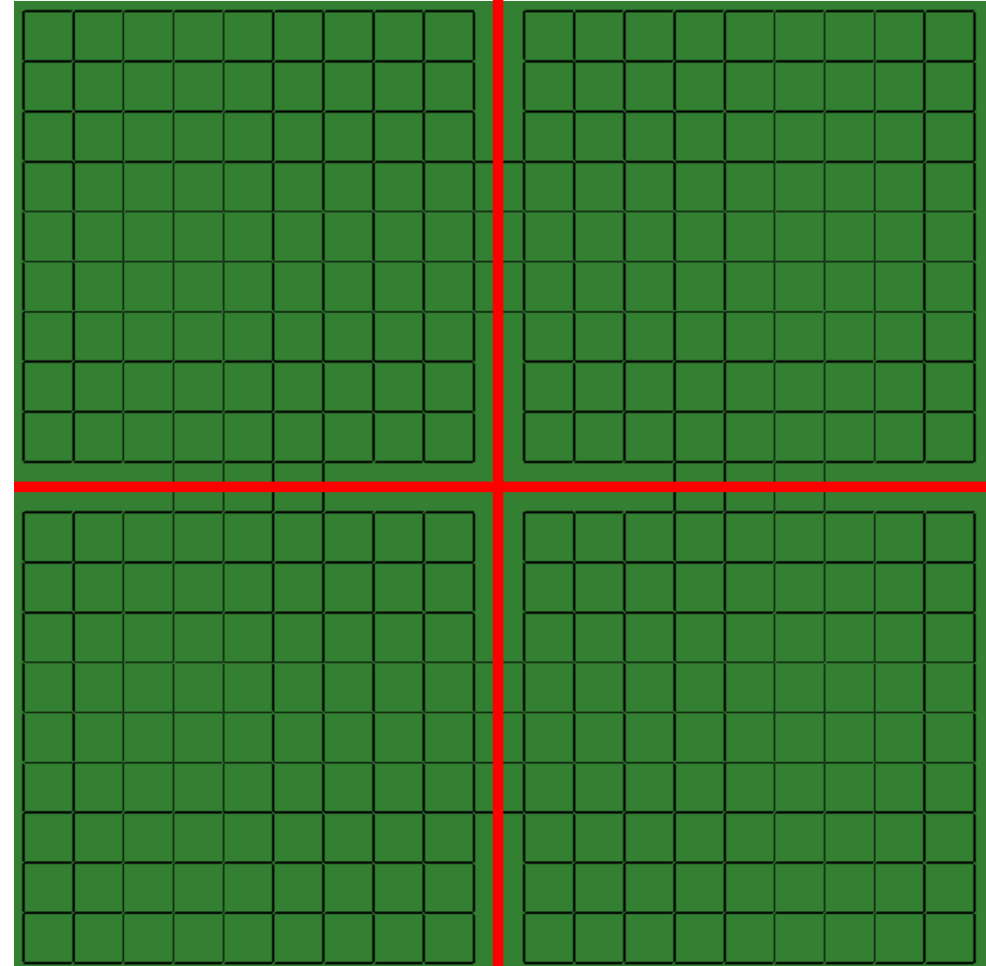
Resultados - escenarios

- Escenario base con 8 enlaces
Comunes y dos particiones



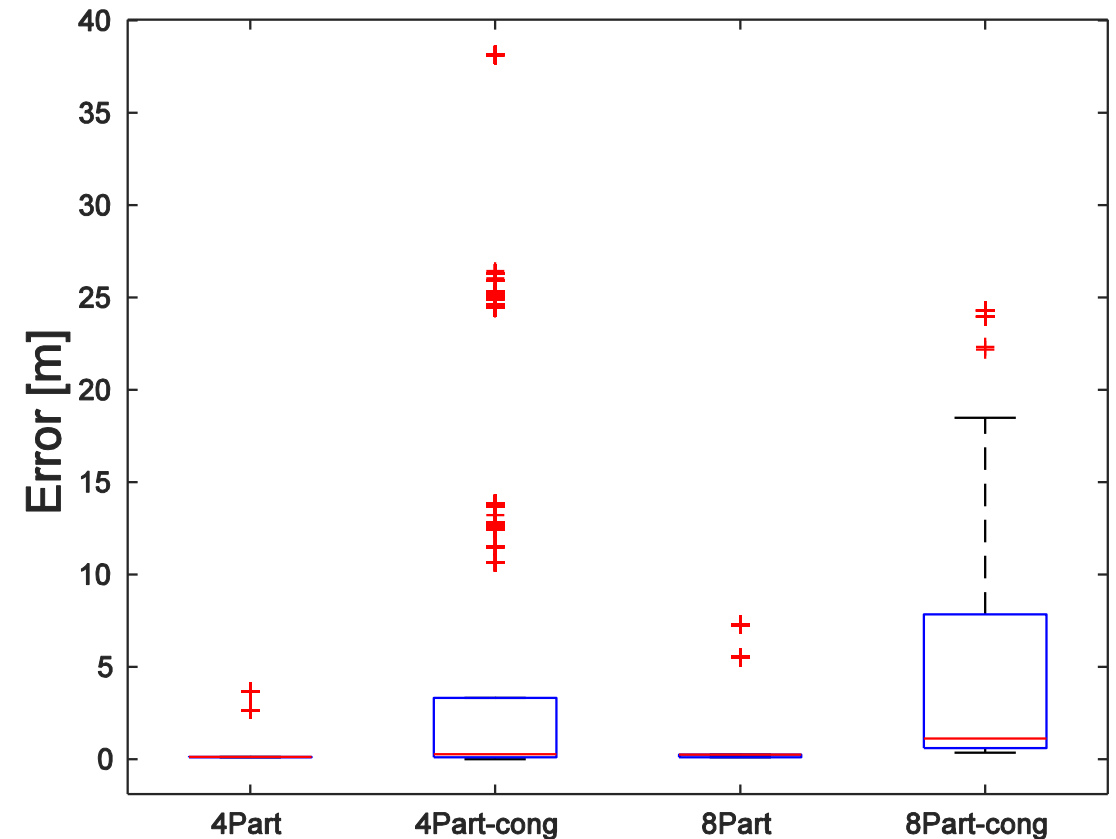
Resultados - escenarios

- Escenario base con 4 particiones y 8 enlaces comunes entre ellos



Resultados - precisión

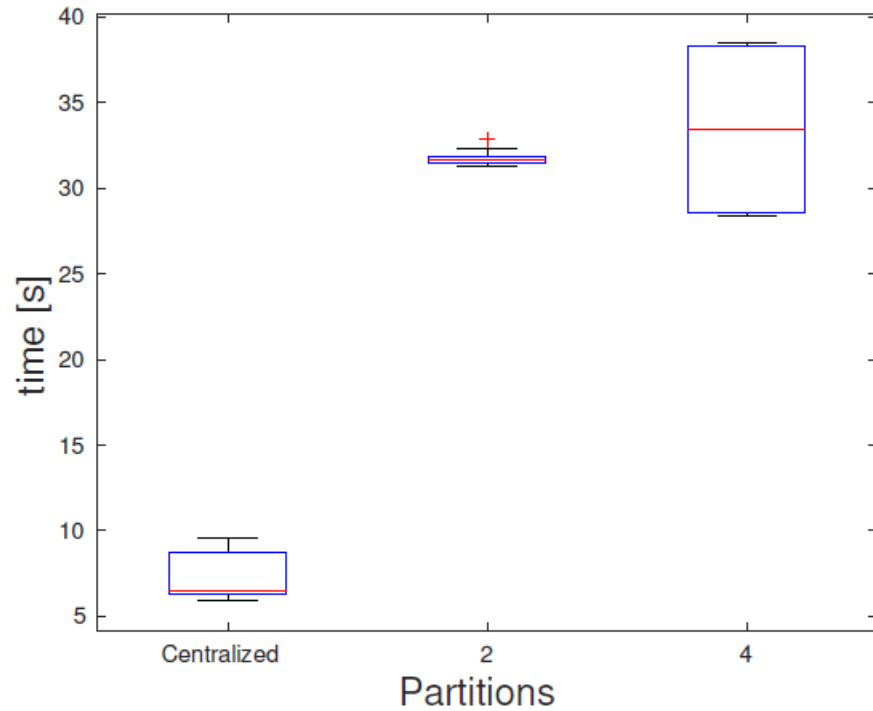
- Precisión:
Diferencia promedio en posición de un vehículo en la simulación distribuida con una simulación normal



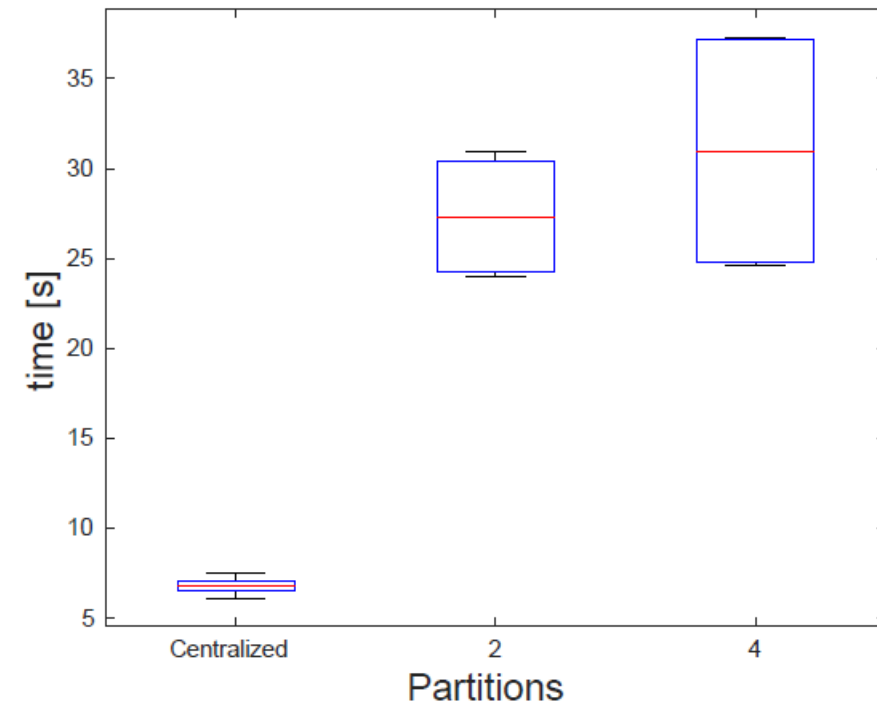
Resultados - precisión

- En los escenarios de flujo libre el error de precisión es prácticamente cero
- En los escenario con congestión los pequeños errores ocasionados por el comportamiento aleatorio de los vehículos se va propagando a lo largo de la ruta
- Estas pequeñas variaciones pueden hacer que vehículos tengan que esperar un ciclo de semáforo adicional, este error se mantiene a lo largo de la ruta

Resultados – desempeño sin ParTraCI



(a) Free flow



(b) Congestion

Resultados – desempeño sin ParTraCI

- Se midió el tiempo que tomó la simulación para los diferentes escenarios tanto en flujo libre como en congestión
- La simulación distribuida no tuvo mejor tiempo que la simulación centralizada

Resultados – desempeño con ParTraCI

- Cuando se usó fuertemente TraCI y ParTraCI el desempeño de la simulación distribuida fue superior
- Se corrió la simulación pero en cada paso de simulación se pedía el listado de los vehículos y para cada vehículo 4 valores numéricos
- Para este caso sólo se usó el escenario de 4 particiones
- Cada simulación se ejecutó 20 veces

Resultados – desempeño con ParTraCI

- Tiempos de simulación promedio en flujo libre para 4 particiones usando TraCI y ParTraCI

| | Flujo Libre | | |
|-----------------|--------------------|-----------------|--------|
| Enlaces comunes | No distribuido [s] | Distribuido [s] | Mejora |
| 4 | 327.6 | 120.5 | 171.9% |
| 8 | 317.0 | 141.4 | 124.1% |
| 40 | 268.0 | 173.8 | 54.2% |

Resultados – desempeño con ParTraCI

- Tiempos de simulación promedio en congestión para 4 particiones usando TraCI y ParTraCI

| | Congestión | | |
|-----------------|--------------------|-----------------|--------|
| Enlaces comunes | No distribuido [s] | Distribuido [s] | Mejora |
| 4 | 354.5 | 130.8 | 171.1% |
| 8 | 346.6 | 137.4 | 152.3% |
| 40 | 1157.9 | 429.4 | 169.7% |

Conclusiones

- El método propuesto tiene una buena precisión y propaga fácilmente cualquier efecto de las particiones maestras
- En desempeño, la simulación distribuida es más lenta que una no distribuida por los procesos de sincronización.
- Cuando ambas simulaciones usan fuertemente TraCI y ParTraCI la simulación distribuida tiene un mejor desempeño
- La implementación de ParTraCI hace que el uso de la simulación distribuida sea prácticamente igual al de una no distribuida



Gracias

Acknowledgement

- Este trabajo fue apoyado por Colciencias mediante el proyecto: Reduccion de Emisiones Vehiculares Mediante el Modelado y Gestion Optima de Traco en Areas Metropolitanas - Caso Medelln - Area Metropolitana del Valle de Aburra, CT 049-2017.



/moycot



@gaunalmed



www.moycot.org