PASA 96

Real-Time Simulation of the German Autobahn Network

April 12th, 1996

Ch. Gawron M. Rickert P. Wagner

email: mr@zpr.uni-koeln.de

http://www.zpr.uni-koeln.de/~mr/research.html

Zentrum für Paralleles Rechnen
Universität zu Köln

Center for Parallel Computing
University of Cologne and
TSASA - Los Alamos National Lab

Talk T17

Real-Time Simulation of the German Autobahn Network

Topics

- CA–Model of Traffic Simulation
- Network Simulation

  Example: Iterative Routing
- Parallelization
- Outlook and Demo

Starting Point

Traffic Simulation

is one of the

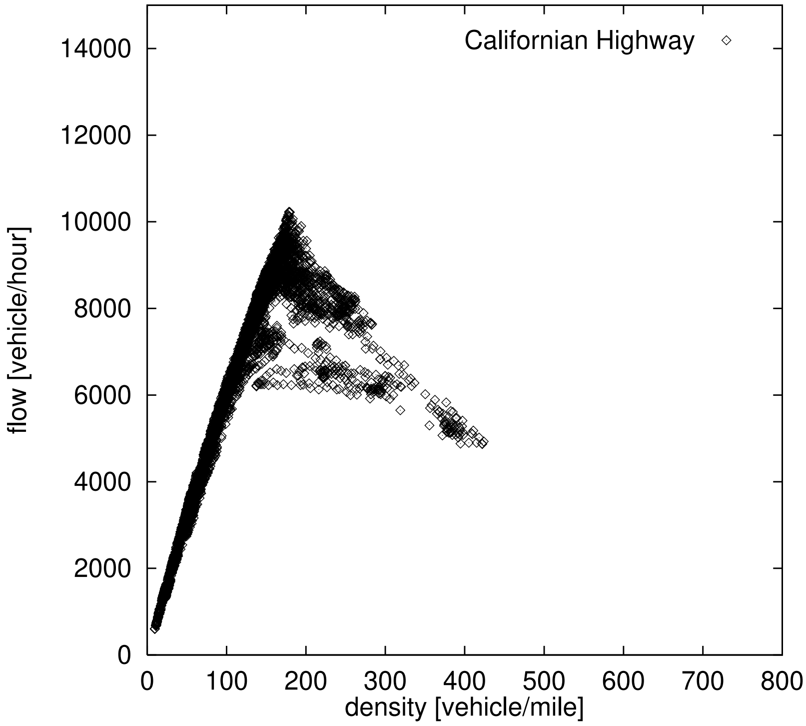
Grand Challenges

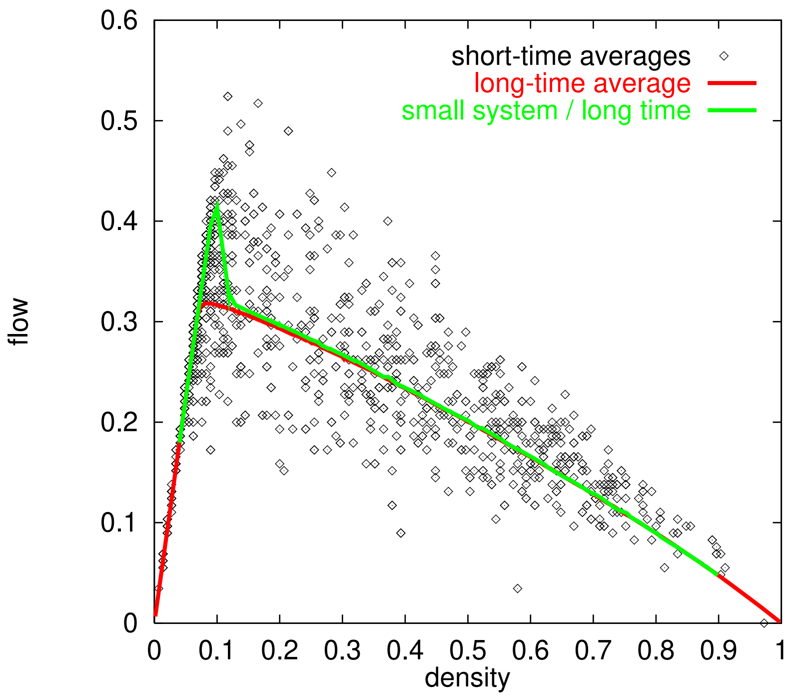
in computer simulation.

Approach

- Fast algorithms based upon
  → Cellular Automata (CA)
- Efficient implementation
  → Parallel Computers

Motivation


Californian Highway
flow [vehicle/hour]
density [vehicle/mile]
0
2000
4000
6000
8000
10000
12000
14000
0
100
200
300
400
500
600
700
800


short-time averages
long-time average
small system / long time
flow
density
0
0.1
0.2
0.3
0.4
0.5
0.6
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Work Groups

TRANSIMS at the Los Alamos Natl. Lab

Los Angelos basin (10 million vehicles)

ZPR Traffic Group in Cologne, Germany

Forschungsverbund Verkehr NRW

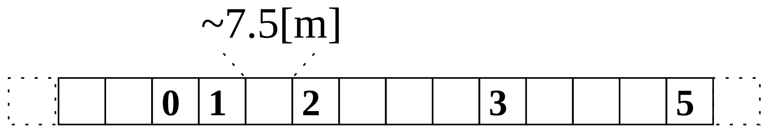
German Autobahn Network

(1 million vehicles)

PARAMICS in Edinburgh, Scottland

Scottish federal road network

Single Lane CA
Nagel / Schreckenberg (1992)



figures denote vehicles with v=0,1,...,5

1 Accelerate:

$$v := \min(v_{max}, v + 1)$$

2 Avoid crash:

$$v := \min(gap, v)$$

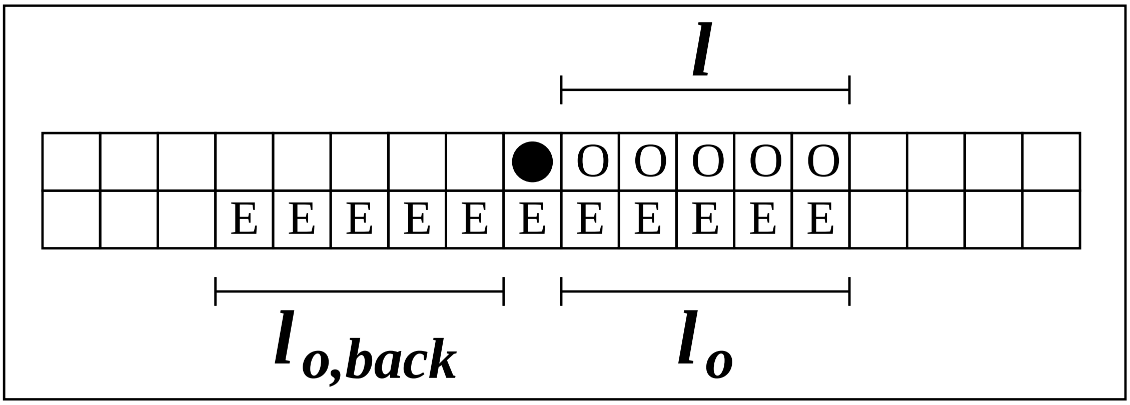
3 Randomize:

$$rand() < p_{dec} \;\Rightarrow\; v := \max(v - 1, 0)$$

Perform Parallel Update

Lane Changing Rules

l	look ahead same lane
l_o	look ahead other lane
$l_{o,back}$	look back other lane

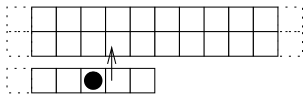
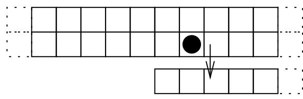
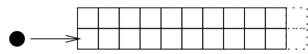
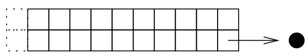


Example

symmetric	asymmetric $L \to R$	asymmetric $R \to L$
$l = v + 1$	no	yes
$l_o = v + 1$	yes	yes
$l_{o,back} = v_{max}$	yes	yes

CA Simulation Network

Building Blocks

- multilane CA
- emission point . . . . 
- absorption point . . . . 
- source . . . . 
- sink . . . . 

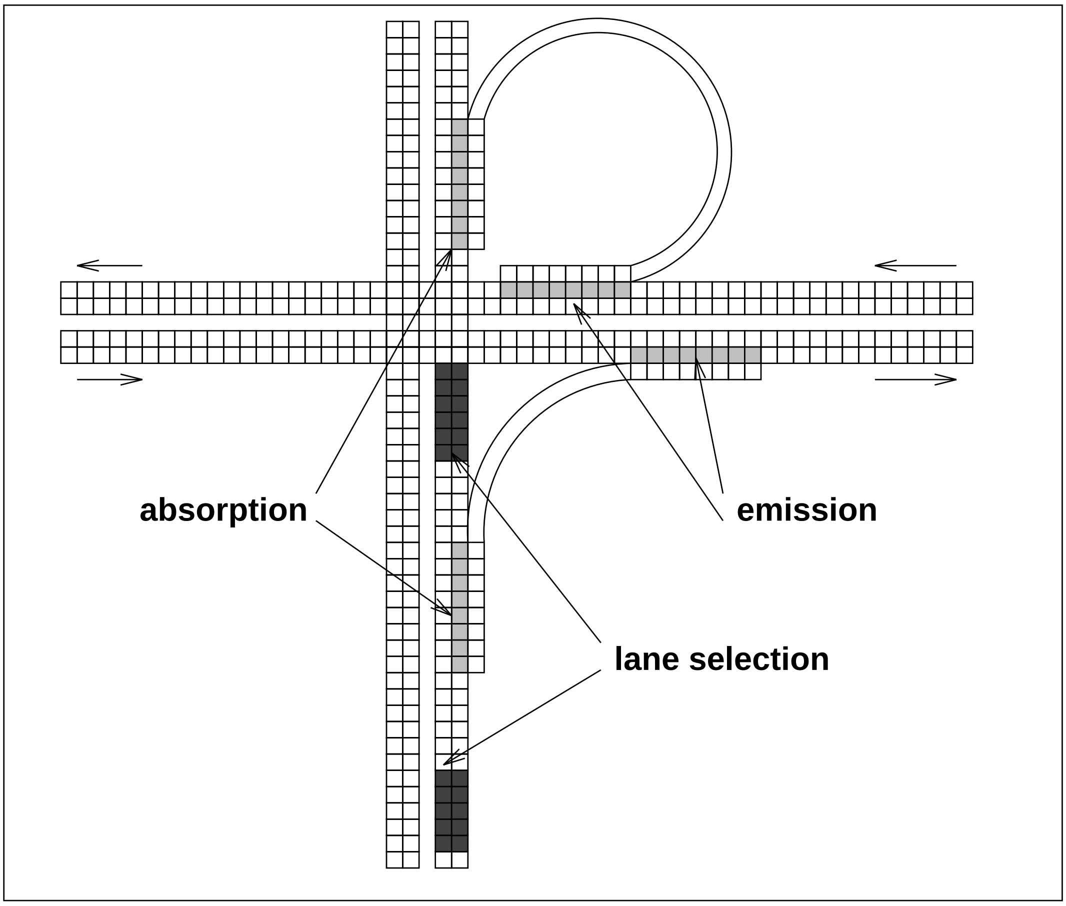
Composite Elements

- net terminator (node degree = 1)
- ramp (node degree = 2)
- intersection (node degree = 3,4)

CA Routeplan Execution

Vehicles

- behave like 'classical' CA on segments
- have individual route plans
- are absorbed/emitted to follow route



Example: Iterative Routing

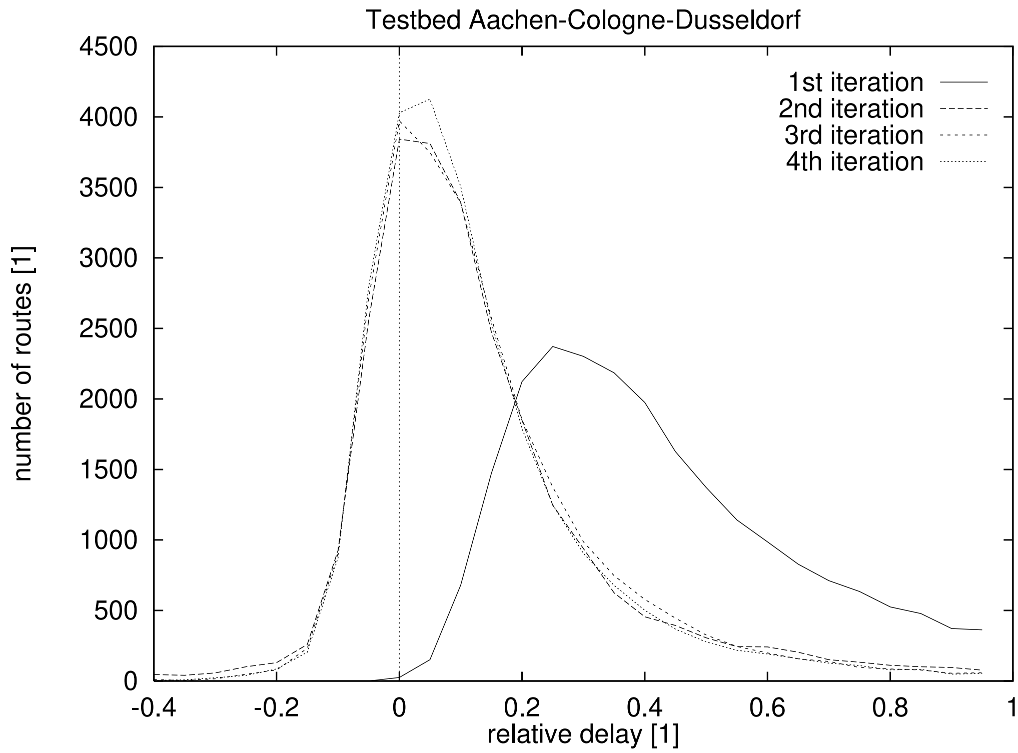
- Input: time-dependent origin-destination matrix
- Output: consistent set of route plans and edge weights

Iteration of Route Planning

1. preload edge weights (e.g. free–flow–velocity)
2. compute route plans for OD-matrix (e.g. shortest paths with Dijkstra)
3. simulate route plans while storing actual time-dependent edge weights
4. goto 2

Routing Example

For low densities ($\varrho = 0.05$) the process converges after the first iteration.



Questions:

Parameter space of convergence?
Quality of prediction?

Parallelization

- traffic network is assigned to a graph of vertices and edges handled by the Parallel Toolbox:

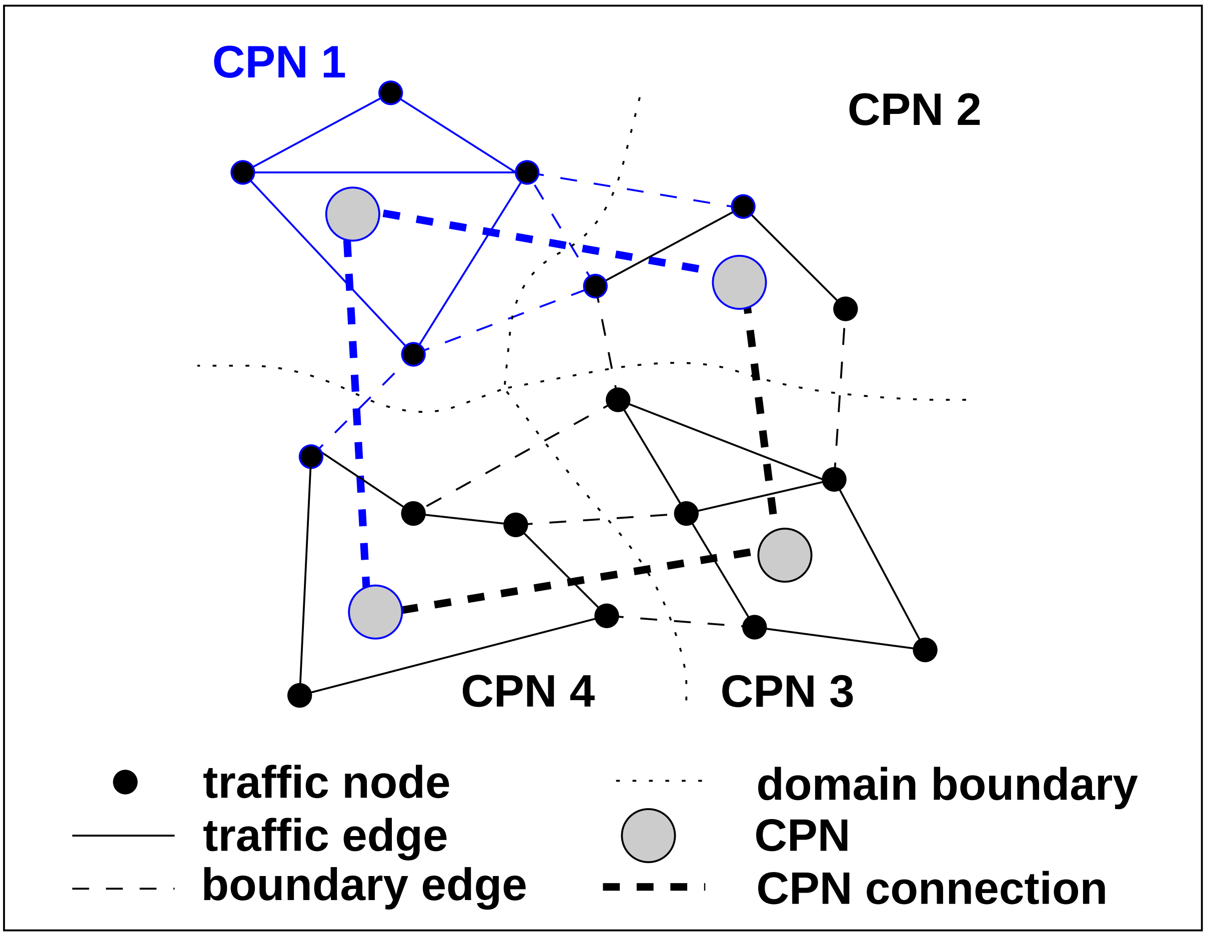
  vertices correspond to terminators, ramps, and intersections

  edges correspond to bidirectional CA multilane segments

- initial geometric distribution of vertices (domain decomposition)
- inter-CPN edges handled by exchange of boundaries
- dynamic load balancing

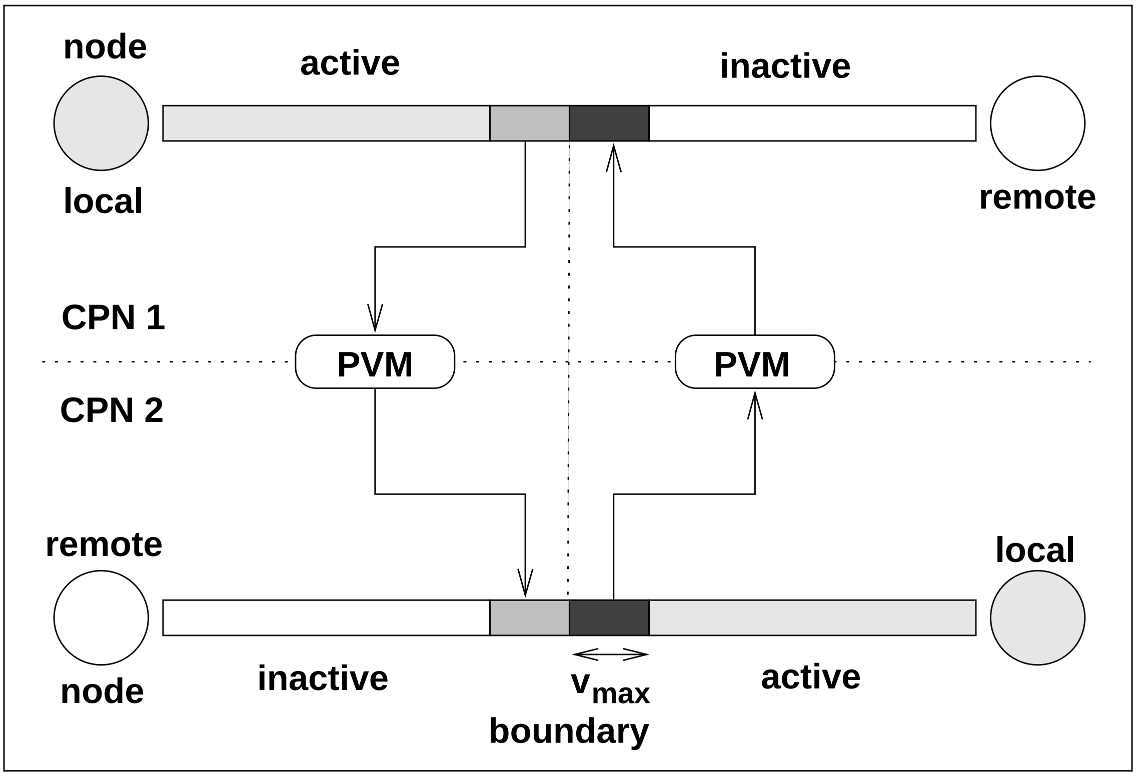
Domain Decomposition

- master CPN has a full copy of inactive network (mainly for graphics)
- each slave CPN has an local active sub network and some inactive dummies



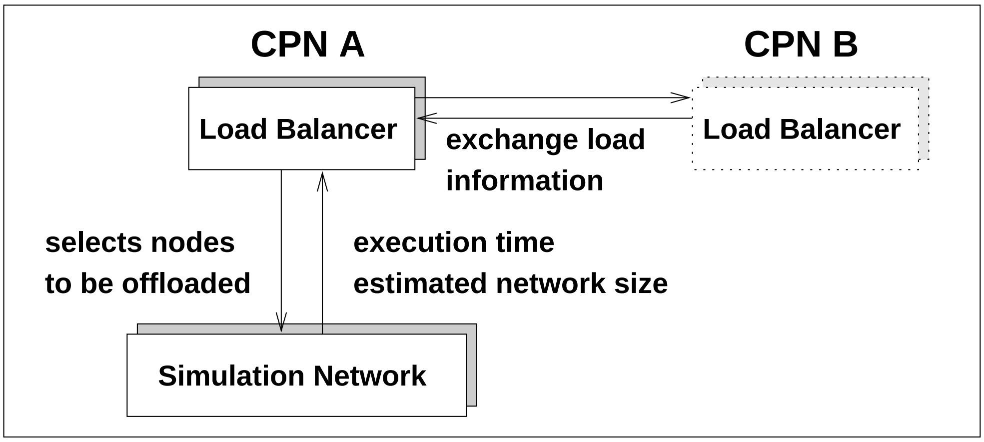
Boundaries

- inter-CPN edges are duplicated with different active ranges
- boundary information is transferred through message passing (PVM)



Load Balancing(I)

Local decision, local migration strategy



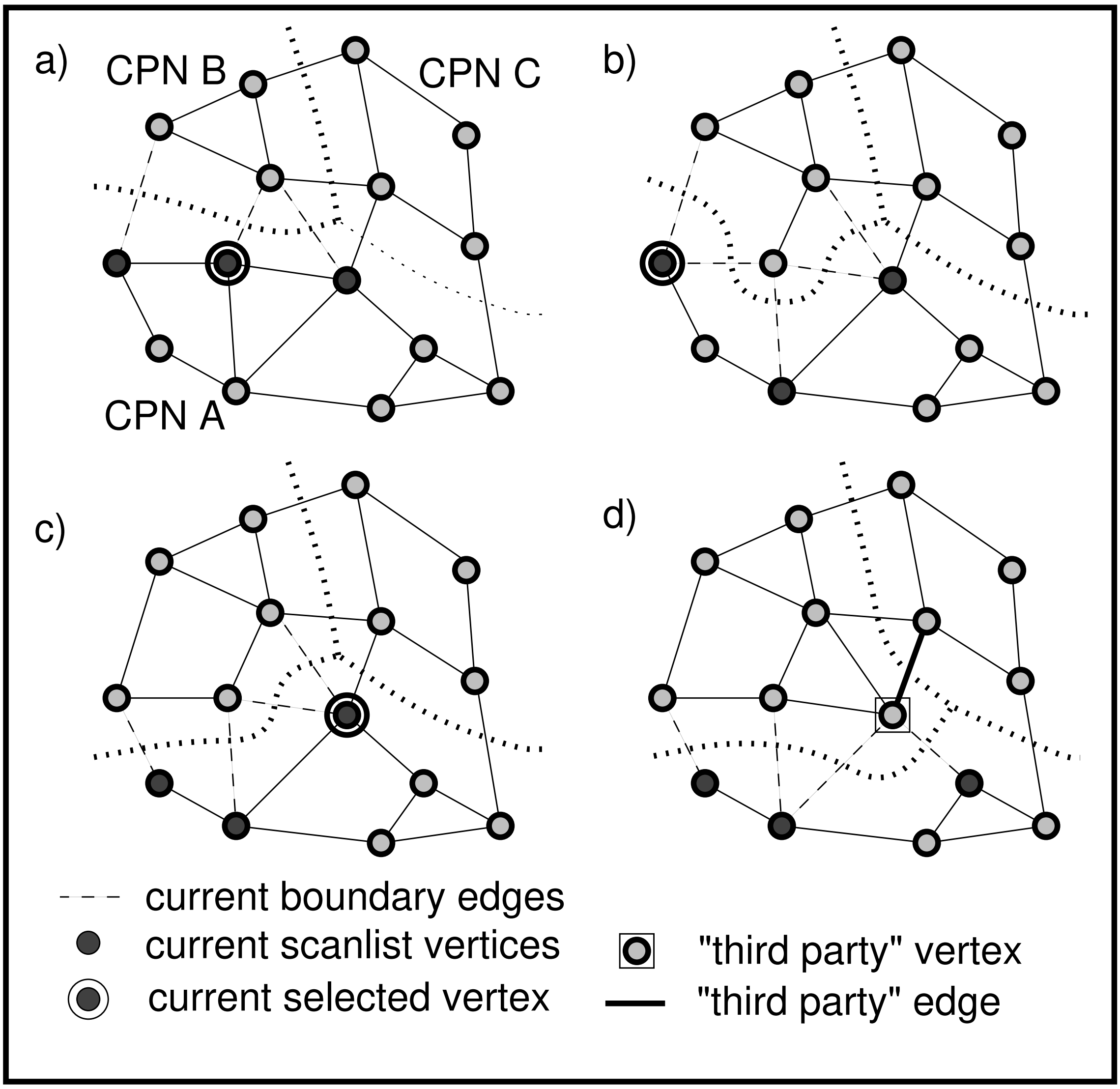
Local vertices are offloaded

- with local synchronization only
- along common boundaries
- preferring vertices furthest from the center (keep 'nice' shape)
- optionally maintaining one connected component per CPN

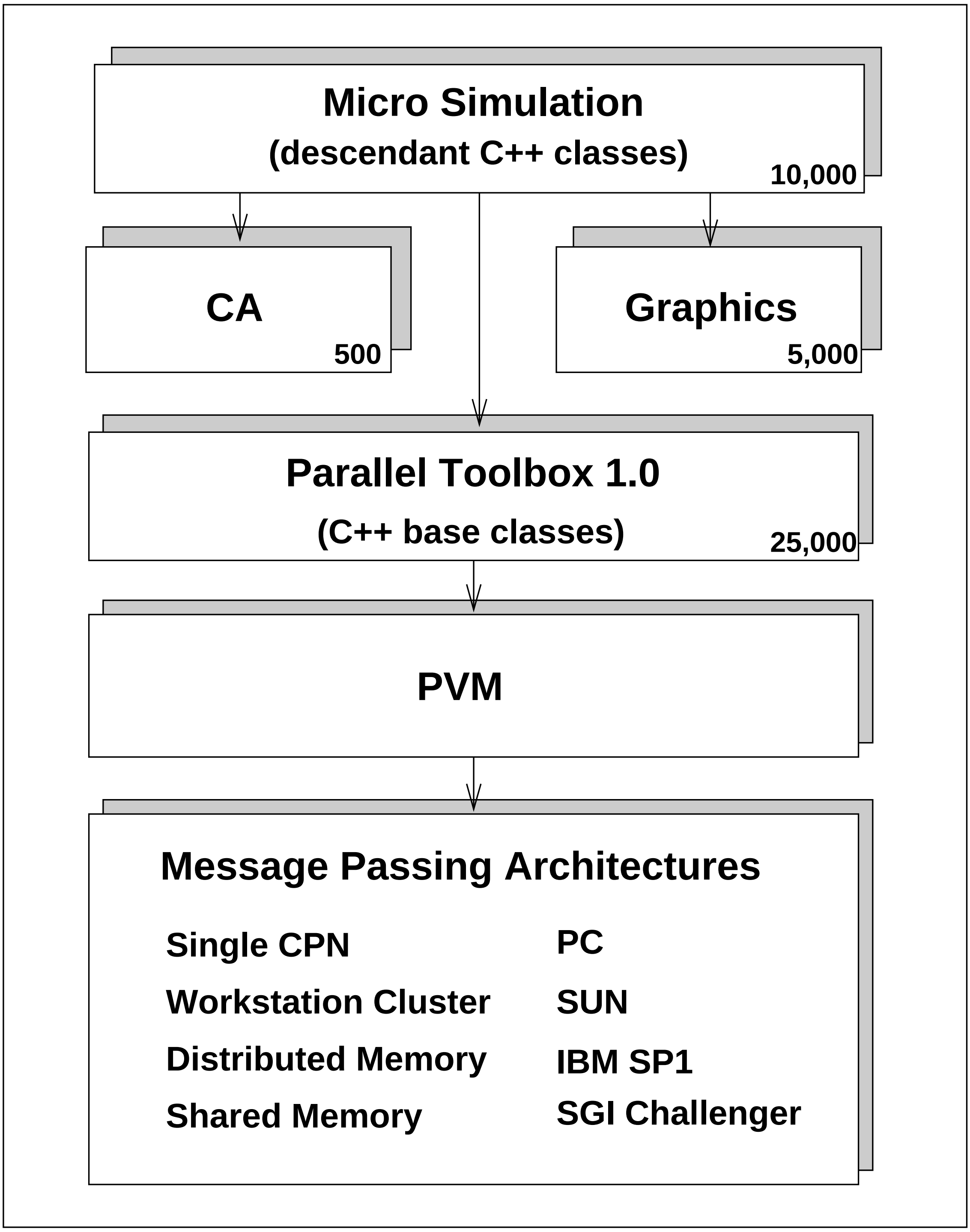
Load Balancing (II)

Offloading vertices along the boundaries

CPN A $\longrightarrow$ CPN B



Current Application Structure



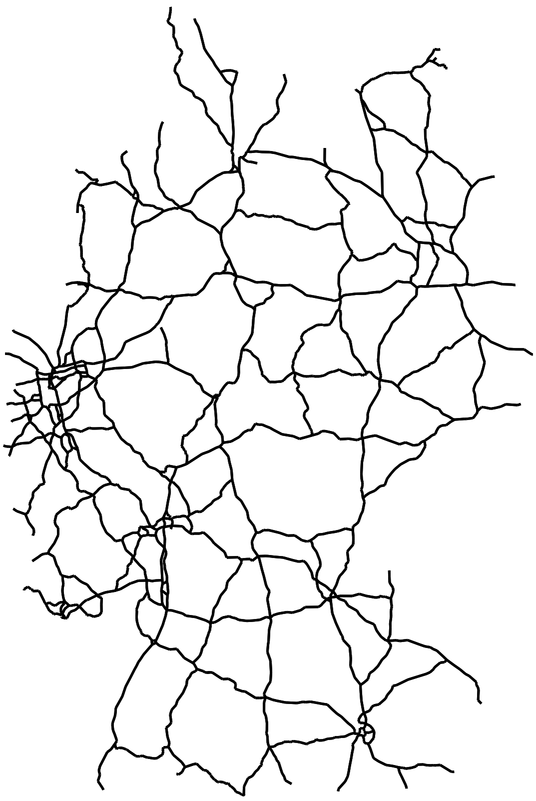
German Autobahn Network

3300 nodes, 3400 edges

$\sim 75,000$ kilometer (lane corrected)

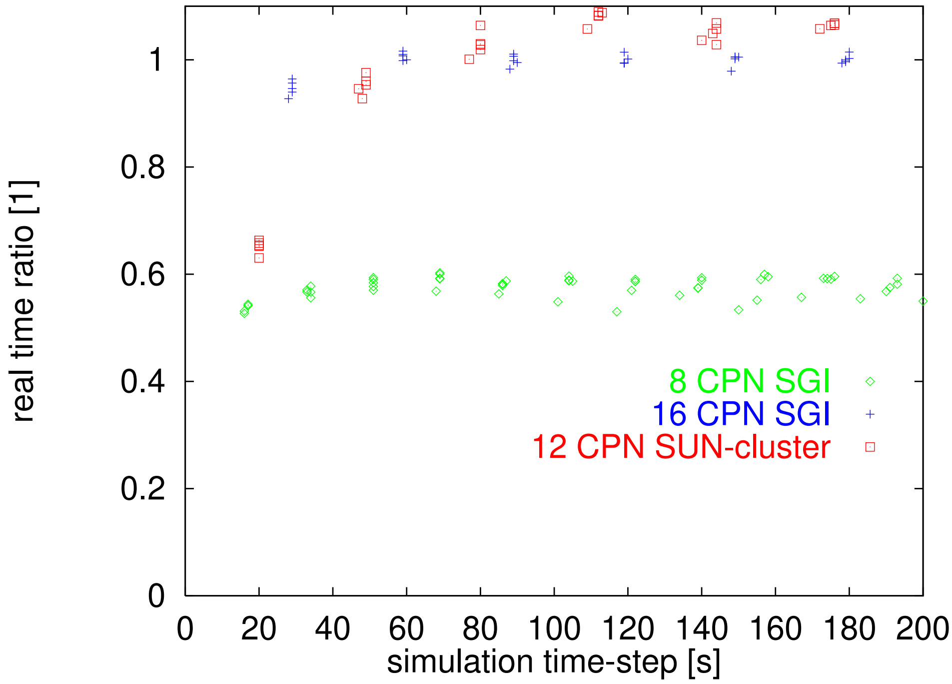
10,000,000 sites

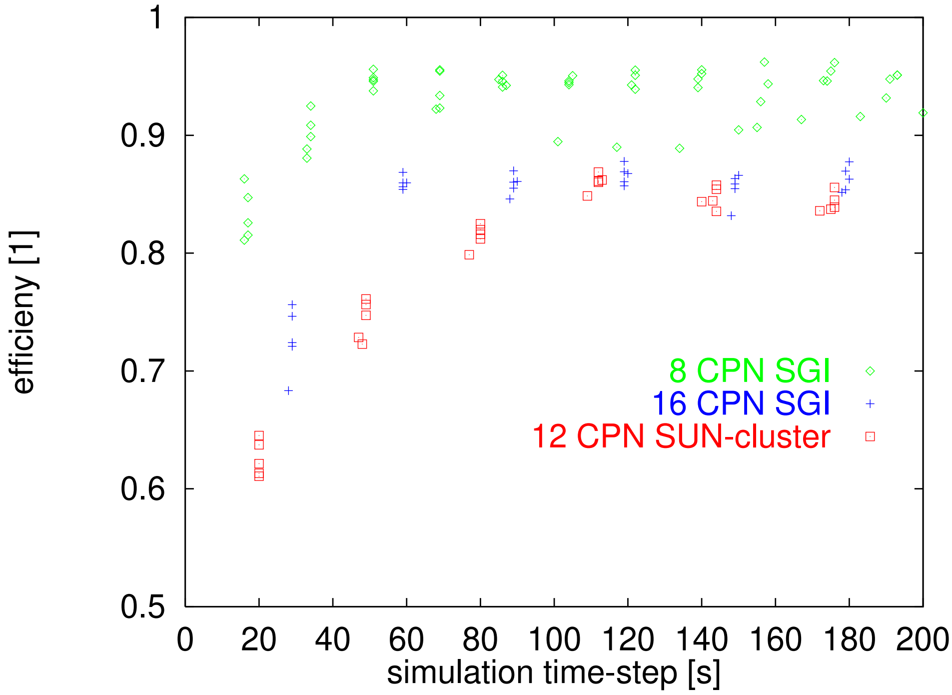
1,000,000 vehicles with routeplans



Network Traffic Simulation

Performance for Map FRG





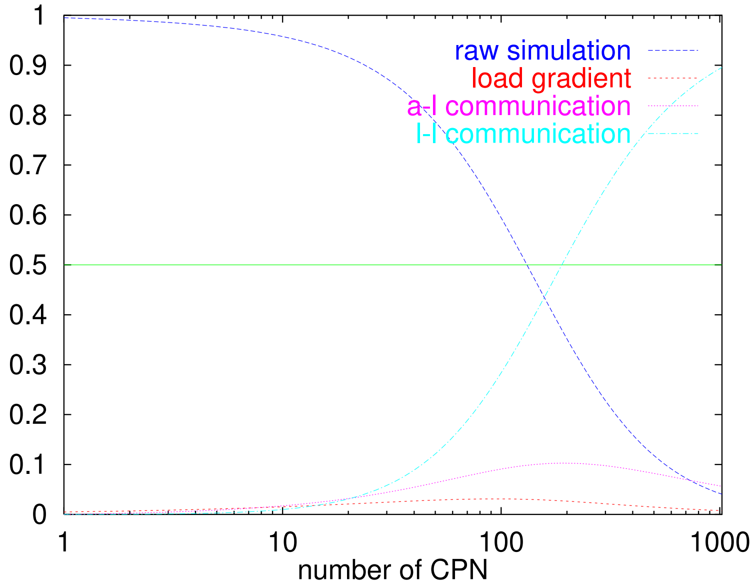
Efficiency Estimates

Criteria

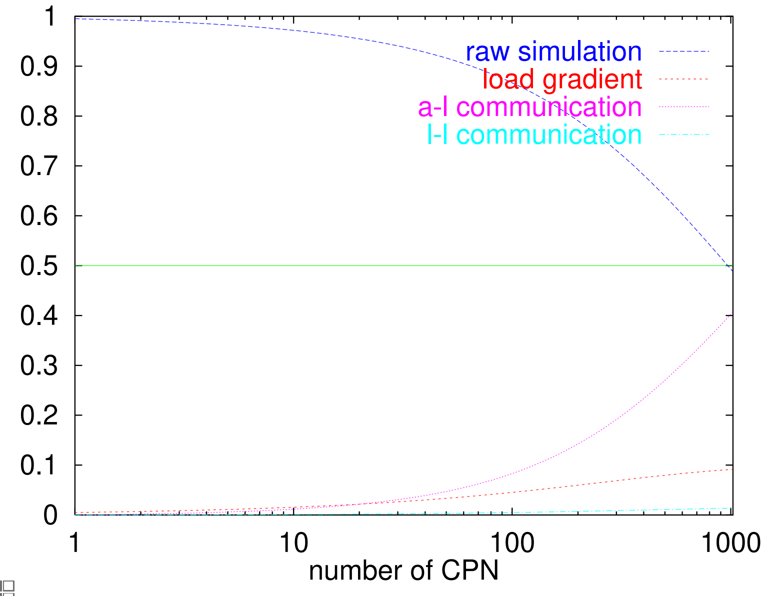
- Application-level communication latency and bandwidth:
  Retrieval, packing, unpacking, and storing of boundary data
- Low-level communication bandwidth:
  Saturation of communication network
- Load gradient:
  Granularity of street network, dynamic load balancing
- Communication topology:
  bus topology or 2-D grid topology

Extrapolated Efficiency

SGI Challenger (bus topology)

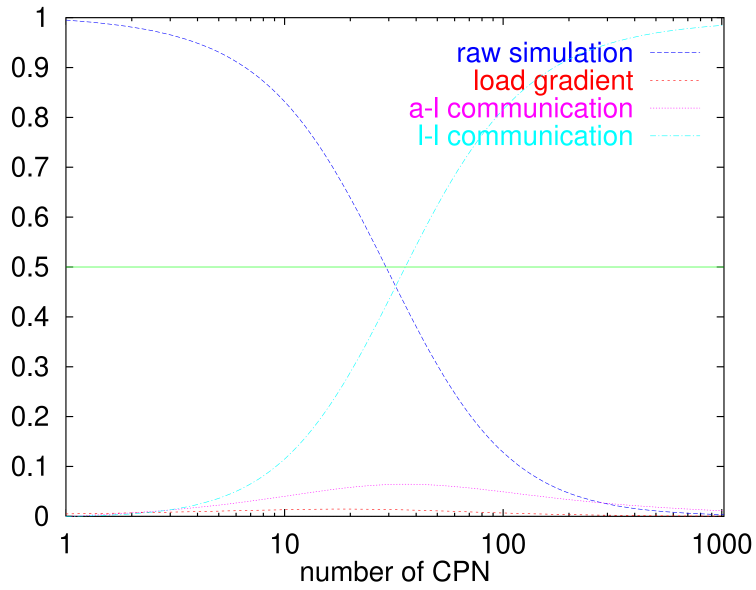


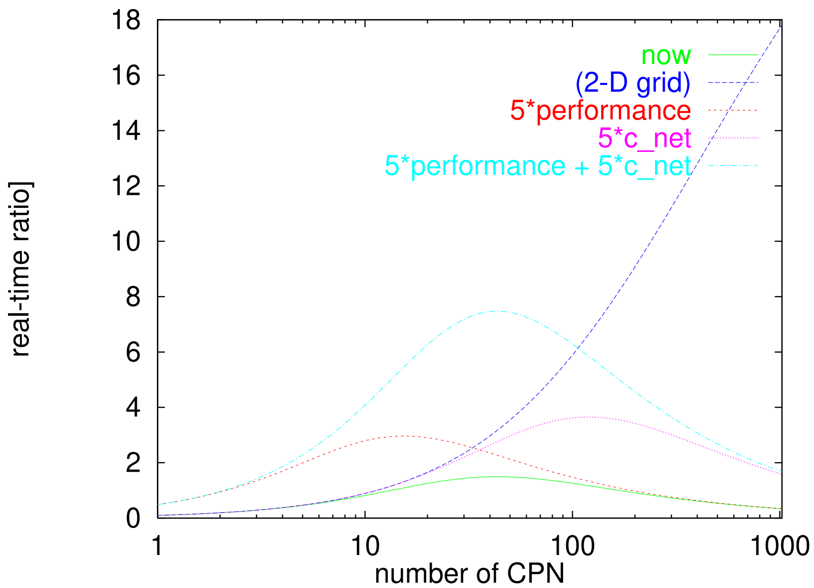
Intel Paragon (2-D grid topology)



Achievable Real-time Ratio with Hardware Tuning

SPARC-5 Workstation Cluster (Ethernet)





Outlook

- Traffic CA
  - include vehicle types
  - produce more realistic multilane fundamental diagrams
  - study net behaviour
- Network Simulation
  - online rerouting
  - examine stability of routing
- Parallelization
  - fewer boundaries
  - global corrections of load gradient
  - 2-D grid topologies

Outlook (II)

- Include Shared Memory to allow for hybrid computer networks