Graph Computing and Its Application in Power System

Guangyi Liu

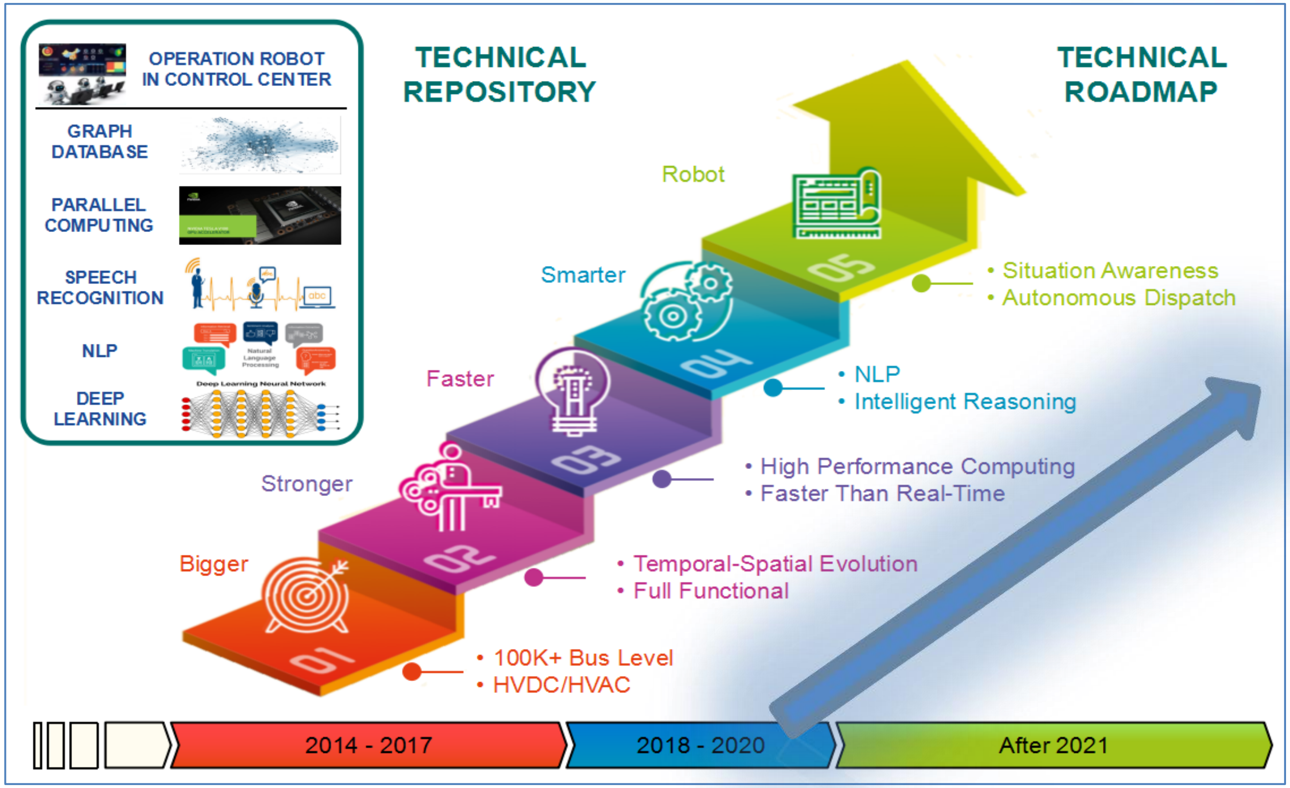
GEIRI North America

January 25, 2019

Content

Next Generation EMS Roadmap

STATE GRID
GEIRI NORTH AMERICA
全球能源互联网美国研究院

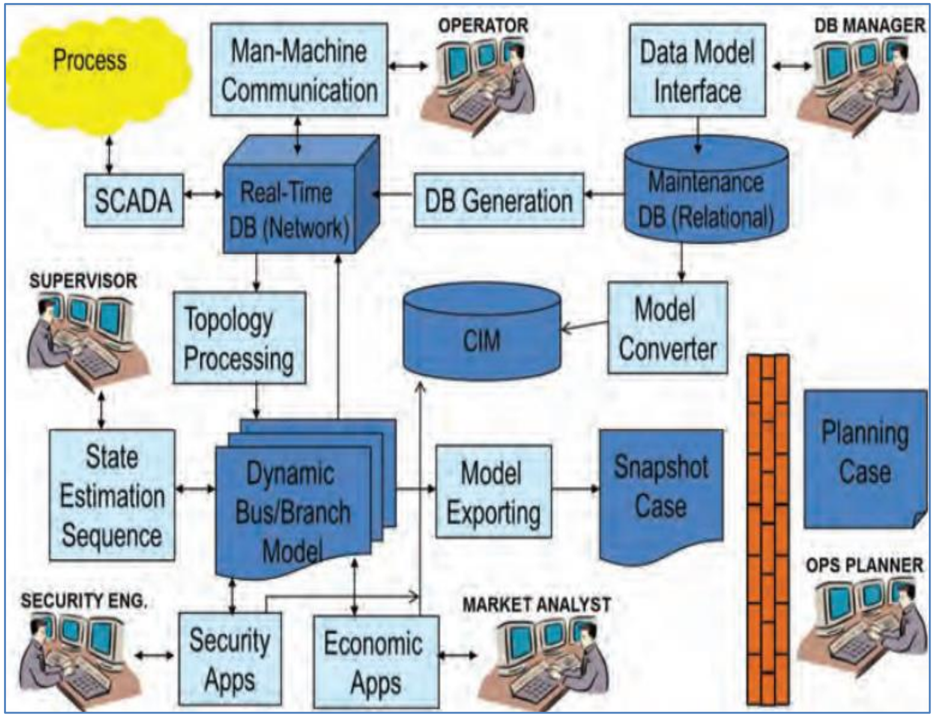


- ❑ 05-The final goal is Robot EMS
- ❑ 02-The industry is at the stage to make EMS full functional
- ❑ 03-The critical path to meet the gap is faster than real-time EMS

Next Generation EMS Goals:

- ✓ Provides real-time, proactive, intelligent, and predictable operation system in control center.
- ✓ Employs graph database, parallel computation, natural language processing, deep learning, and situation awareness and autonomous dispatch to drive anlaytical EMS to intelligent/robot EMS.

Super-fast EMS
Why We Need it



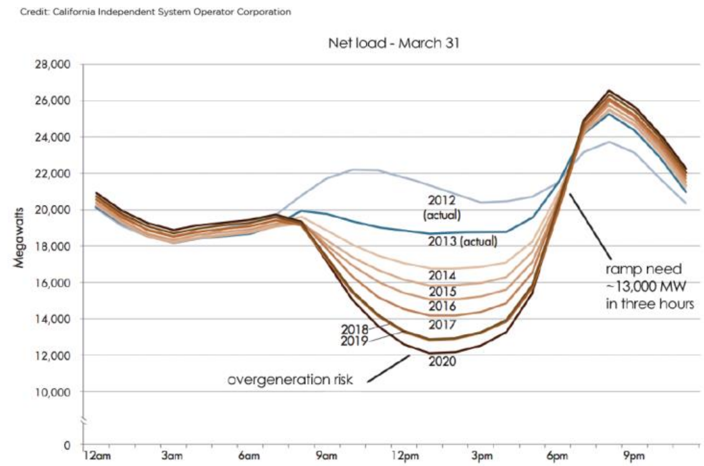
EMS at Present **Gap** **EMS in Future**

- EMS in future supported by AI decision requires sophisticated model, intensive calculation, and **fast computing**
- **High performance computing** is the key to make EMS intelligence
- **Accelerating** application computing is critical for next generation EMS

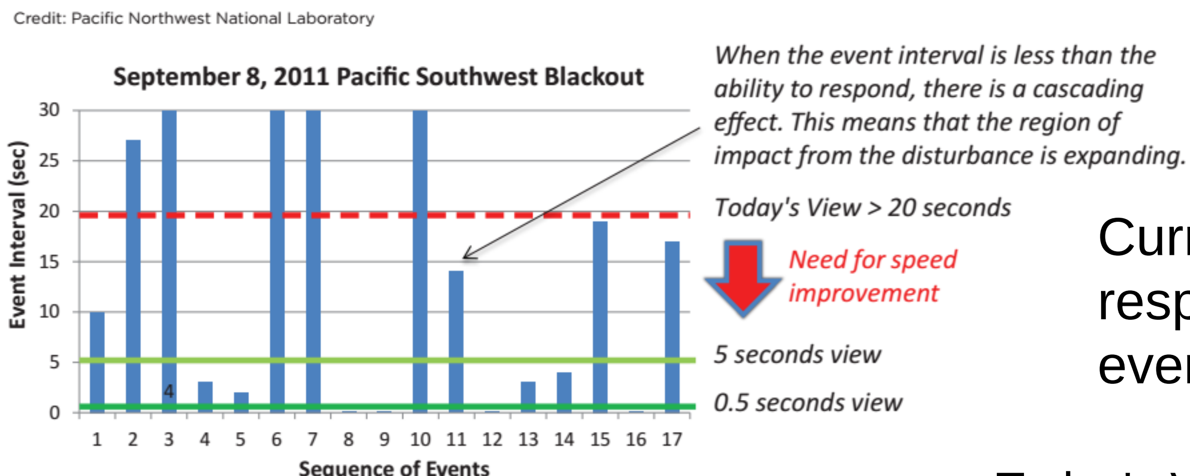
Super-fast EMS

Why We Need it

Power Electronics Dominated Large and Complex System



Fast Change of Loads in Minutes Due to the Renewable Intermittence

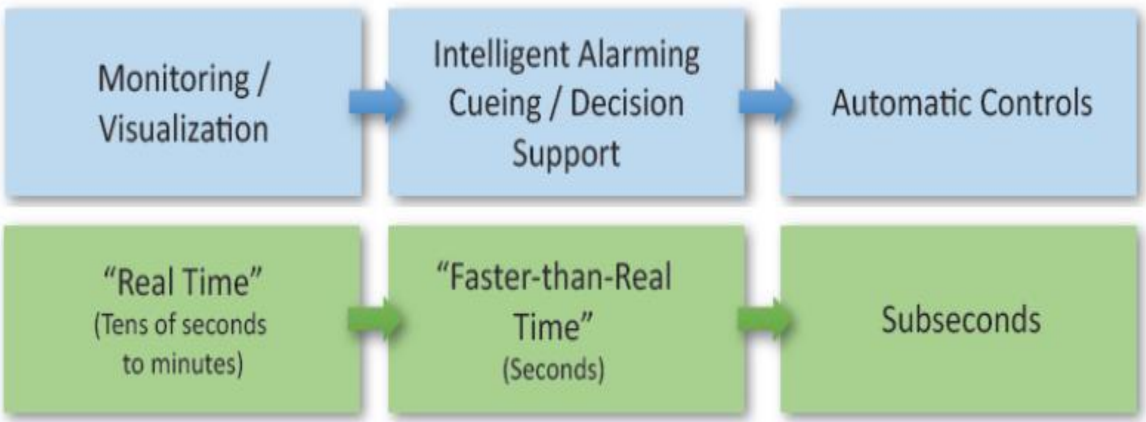


Sequence of Cascading Events in the 2011 Southwest Blackout in the US

Current EMS cycle delays responses to the cascding events driving the blackout

Today's View > 20 seconds
Needed View < 0.5 seconds

Super-fast EMS
Why We Need it



Pathway to Speed Improvements in Analytical Decision Making

Subseconds analytical processing time enabling the critical functions:

- ❑ Security Constrained Automatic Generation Controls
- ❑ On-line SPS Arming Decision
- ❑ Heavy Loaded Power Flow Accurate Solution

- ❑ The analytical processing time needs to be reduced, from tens of seconds to subseconds, to move from monitoring and visualization to automatic controls.
- ❑ The need for fast and predictive analytics is amplified by physical and cyber attack on critical infrastructure.
- ❑ US DOE requires to develop State Estimation at 0.5 seconds speed for medium size system. DOE funded $220 Millions to Faster Real-time Analytical Tools.

Super-fast EMS
What We Achieved

Test System： A Real Provincial System （2650 Bus）			
Commerical EMS		Faster-than-Real-Time EMS （GEIRINA EMS Prototype）	
SCADA Sampling Rate	***5*** s	SCADA Sampling Rate	***5*** s
EMS Execution Cycle	**60-300 s**	EMS Execution Cycle	**5 s**
SE Execution Time	**~4490 ms**	SE Execution Time	**~200 ms**
PF Execution Time	**~3820 ms**	PF Execution Time	**~70 ms**
CA Execution Time	**~18000 ms**	CA Execution Time	**~1300 ms**

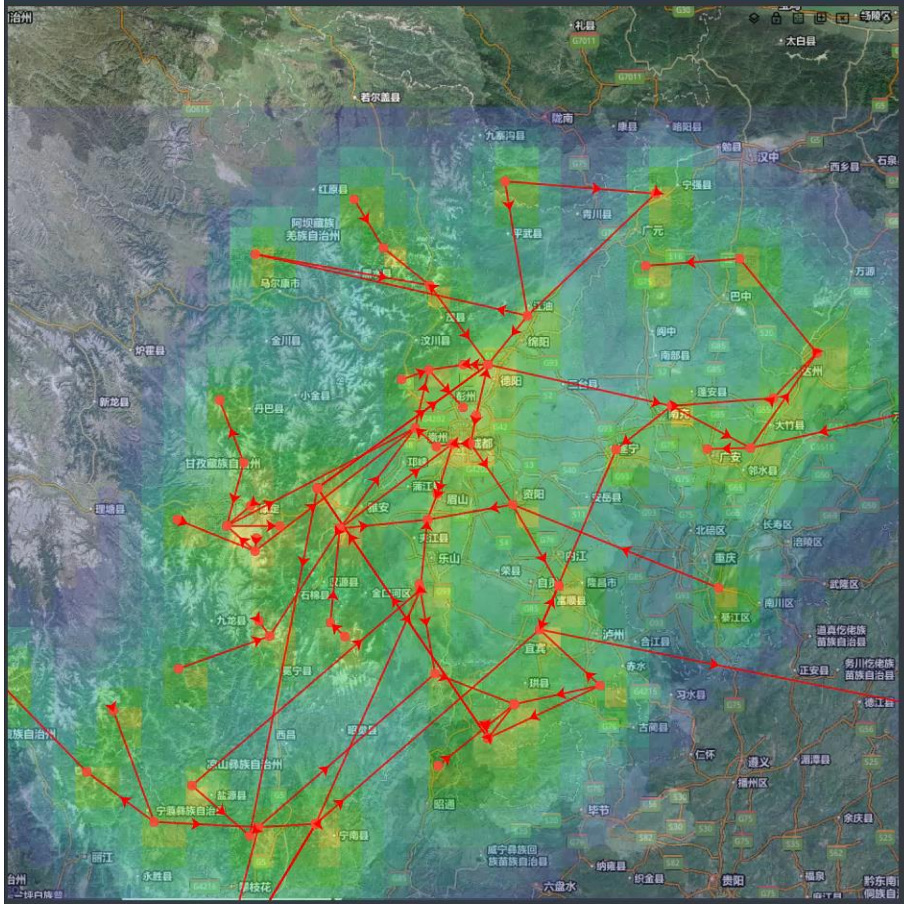
Source: Commerical EMS

Source: GEIRINA EMS Prototype

Super-fast EMS

What We Achieved - Demo

超实时仿真EMS系统 状态估计 在线潮流 安全分析



系统概况

节点数:	2687	支路数:	3208
SCADA采样时间:	15:01:53.877	EMS计算周期:	5s
平衡母线:	四川.蜀州/500kV.I母线		

拓扑分析

运行状态:	完成	运行时间:	3ms
起始时间:	15:01:53.881	结束时间:	15:01:53.884
遥测更新数据:	0	遥信更新数据:	0

状态估计

运行状态:	完成	运行时间:	139ms
起始时间:	15:01:53.887	结束时间:	15:01:54.026
状态估计算法:	加权最小二乘法	迭代精度:	0.001
迭代次数:	5		

在线潮流

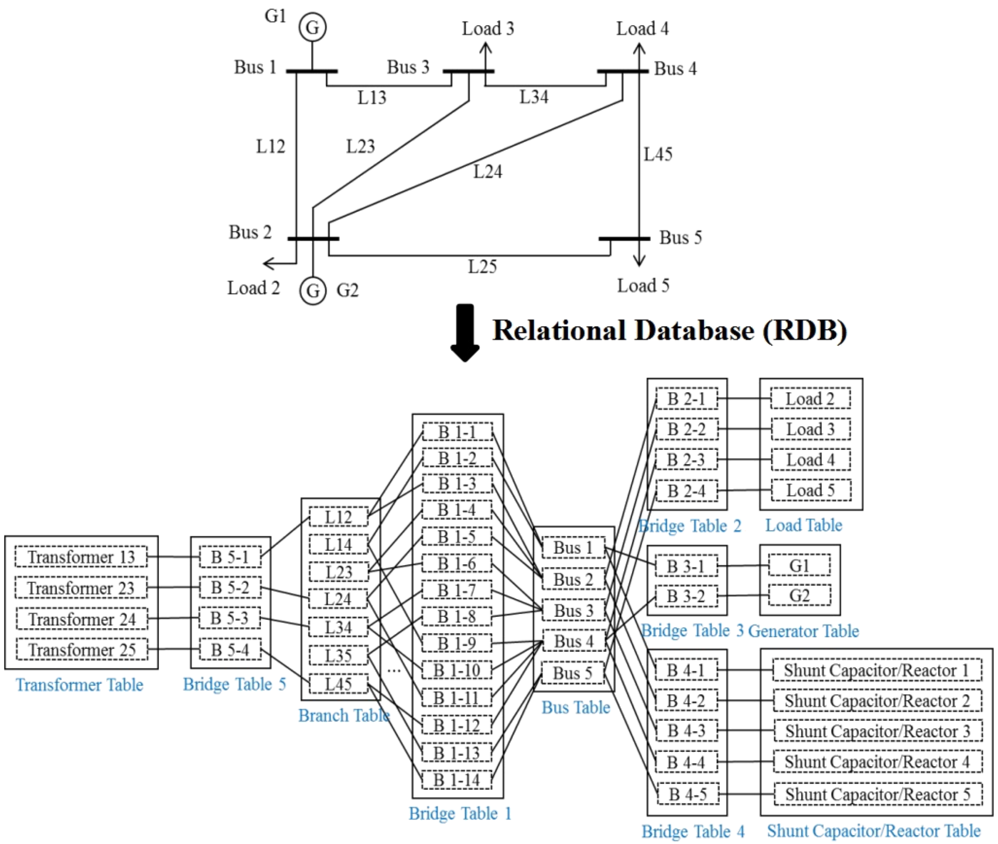
运行状态:	完成	运行时间:	56ms
起始时间:	15:01:54.029	结束时间:	15:01:54.085
潮流算法:	P-Q分解法	迭代精度:	0.05
迭代次数:	5		

安全分析

运行状态:	完成	运行时间:	1322ms
起始时间:	15:01:54.087	结束时间:	15:01:50.418
快速扫描样本数:	1093	安全分析算法:	叠加法

Please note the running-time on the prototype is wall clock time incluing function calling time, execution time, and communication time.

RDB for Power System Modeling



Physical System

- ❑ Nodes are connected by edges
- ❑ Connectivity is naturally a graph

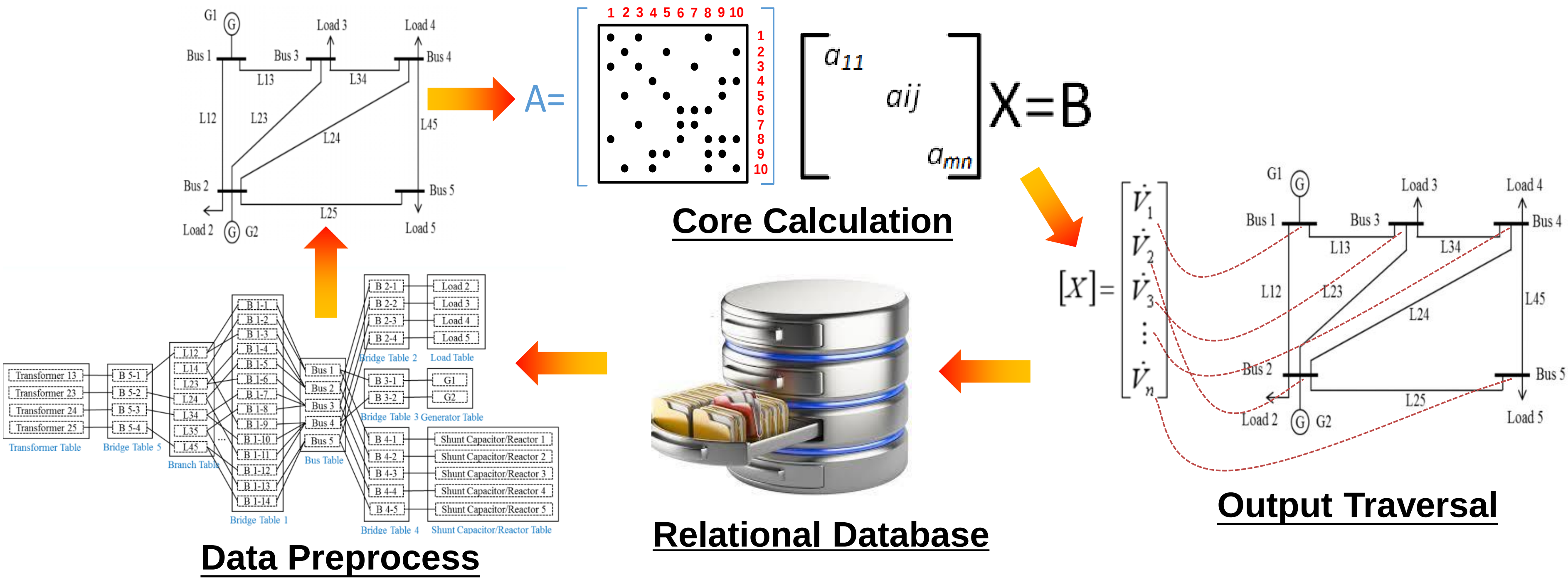
Relational Database

- ❑ Use table structure
- ❑ Not support unstructured data
- ❑ Attribute relations modelled by separated tables
- ❑ Use commonly shared key values to represent data relationships

Issues of Relational Database for Power System Modeling

- ❑ Join intensive queries for the whole database invite large computation time
- ❑ Maintain small portion of system requires multiple table update
- ❑ Time consuming to support recursive queries and parallel queries

RDB based Power System Computing

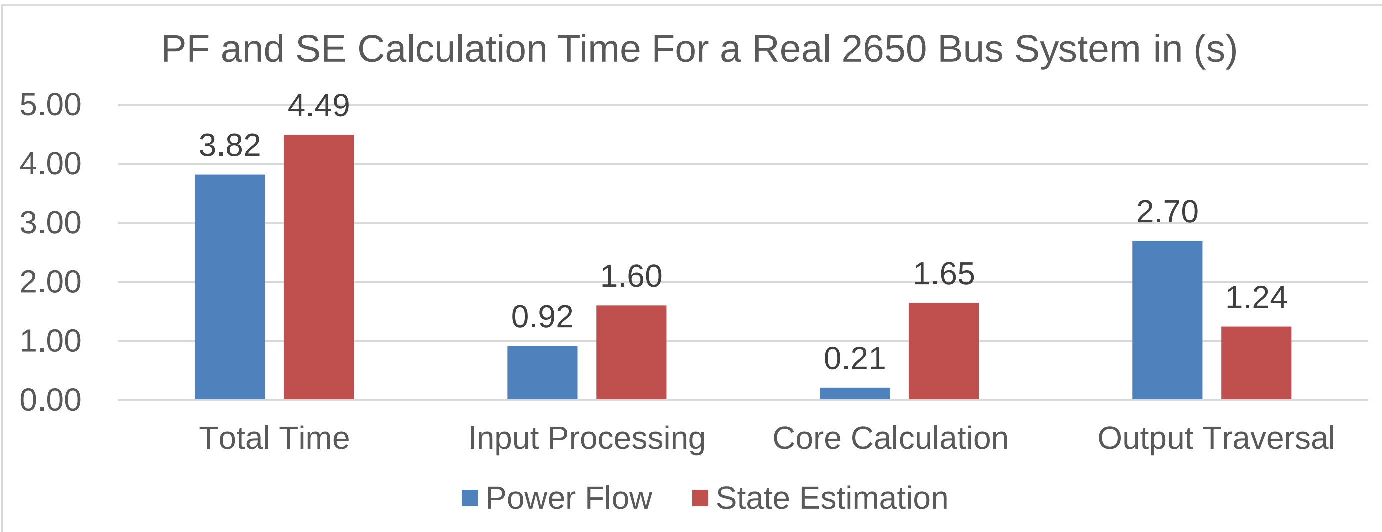


Issues of Relational Database for Power System Computing

- ❑ Need loop through branch table and bus table to create connectivity
- ❑ Complicated to support linear equation parallel computing
- ❑ Map solved variables to bus voltages and branch flows inviting time consuming output traversal

RDB Time Decomposition

A Real Case of 2650 bus system

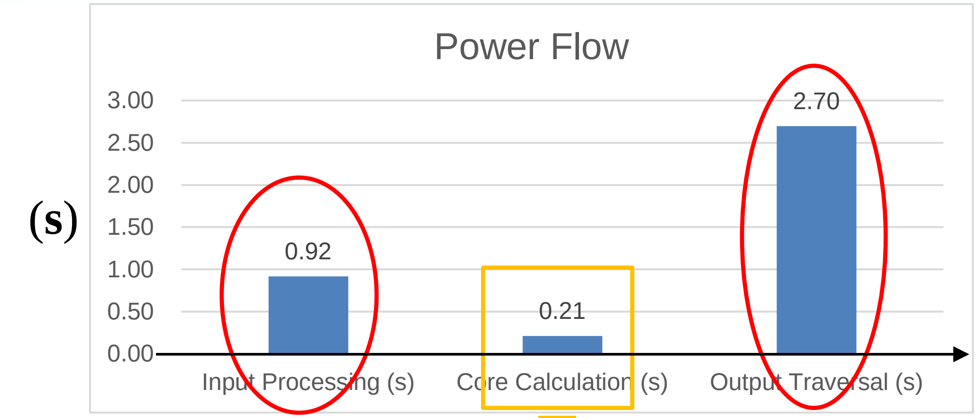


Data source: Beijing Kedong Electric Power Control System Co., LTD.

Observations:

- ❑ PF Data Input Processing and Output Traversal cost 94.5% of the total time
- ❑ SE Data Input Processing and Output Traversal cost 64.3% of the total time
- ❑ Need a new platform to integrate data management, calculation, and visualization

Graph Potentials
- Parallel Computing - PF

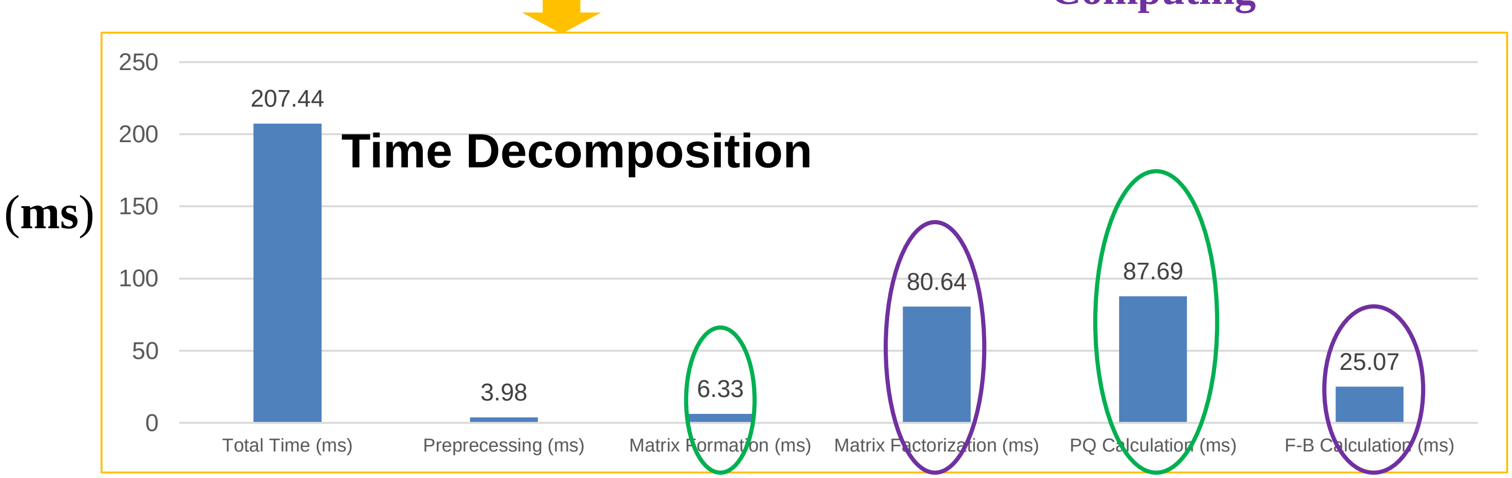


Using graph database model to save the time of

- ❑ **Input Processing**
- ❑ **Ouptut Traversal**

Further improve Core Calculation efficiency by

- ❑ **Graph based Nodal Parallel Computing**
- ❑ **Parallel LU Solver Hierarchical Parallel Computing**

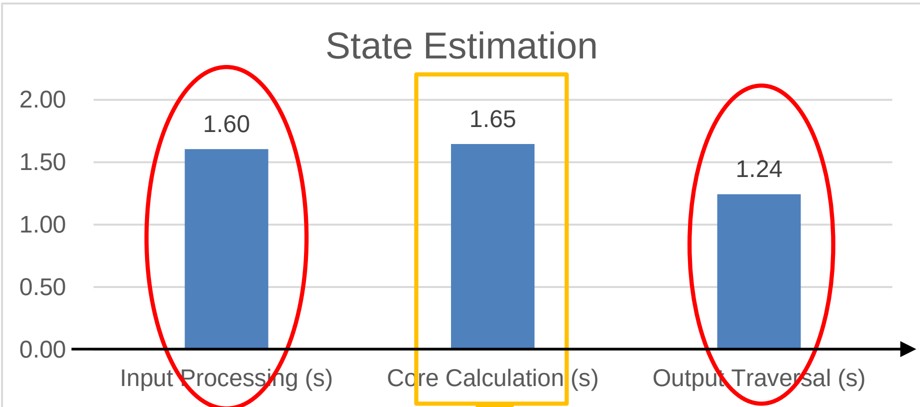


Observations:

- ❑ Matrix formation and P/Q calculation cost 45% of the total time which can be nodal parallelized.
- ❑ Matrix factorization and F/B substitution take 51% of the total time which can be hierarchical parallelized.

Graph Potentials
- Parallel Computing - SE

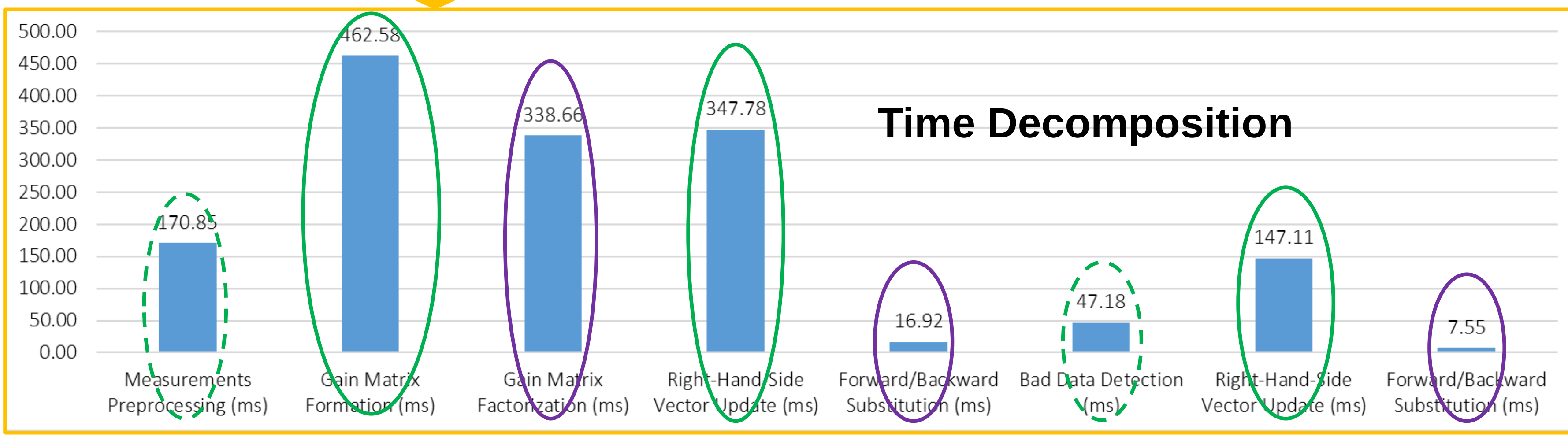
(s)



Using graph database model to save the time of
- ❑ Input Processing
- ❑ Ouptut Traversal

Further improve Core Calculation efficiency by
- ❑ Graph based Nodal Parallel Computing
- ❑ Parallel LU Solver Hierarchical Parallel Computing

(ms)

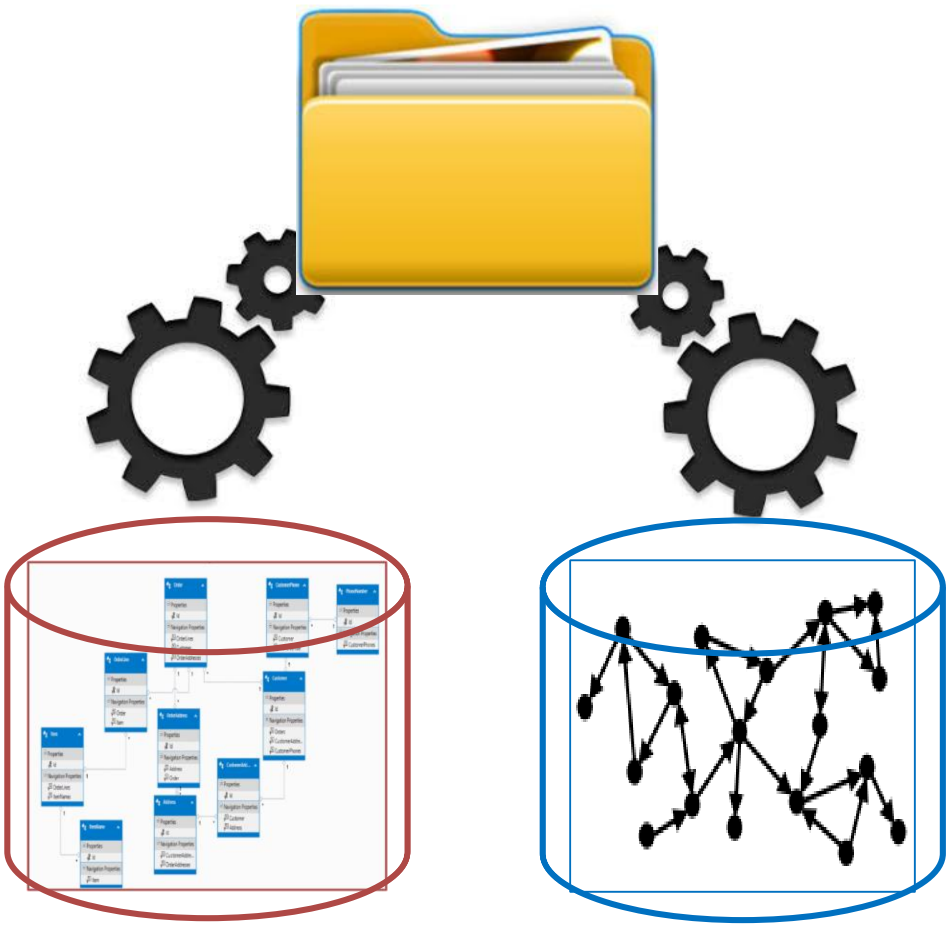


Observations:

- ❑ Gain matrix formulation and right-hand-side vector update take ~60% of core computation time which can be implemented by nodal parallel computing.
- ❑ Gain-matrix factorization and forward/backward substitution cost ~40% of time which can be hierarchically parallelized.

GDB for Power System Modeling

Original Data Document



Relational Database:
Data model is a collection of interlinked tables.

Graph Database:
Data model is a multi-relational graph.

Relational Database

- ❑ Use table structure
- ❑ Attribute relations modelled by separated tables
- ❑ Need to update multiple table to maintain system
- ❑ Hard to support recursive queries and parallel queries

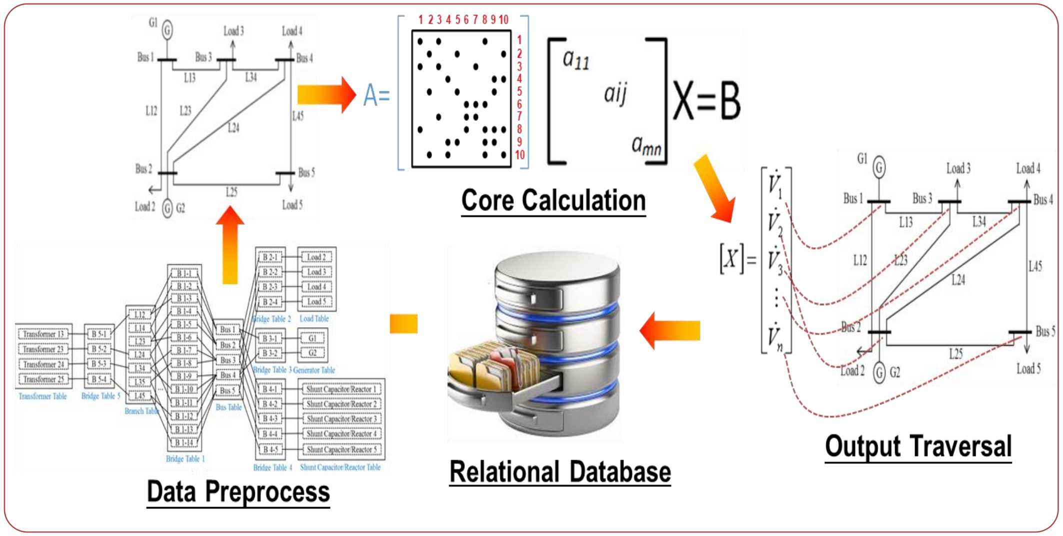
Physical System

- ❑ Edges are connecting by nodes
- ❑ Connectivity is naturally a graph

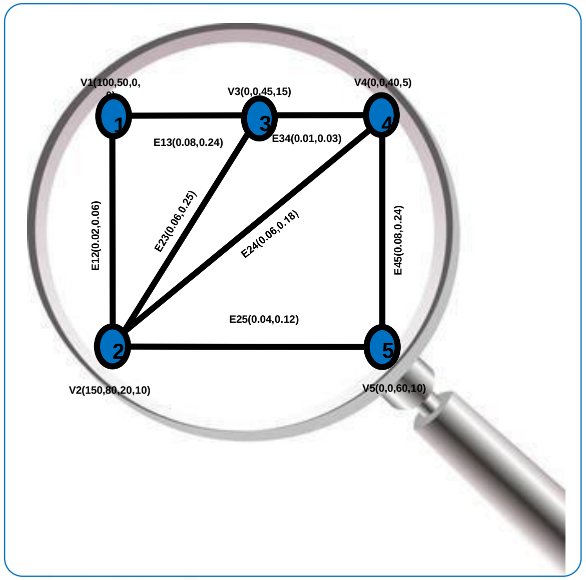
Graph Database

- ❑ Use graph structure with edges and nodes
- ❑ Store data by attributes of nodes and edges
- ❑ Support parallel computing
- ❑ Easy to maintain large system

GDB based Power System Computing



RDB Based Computing

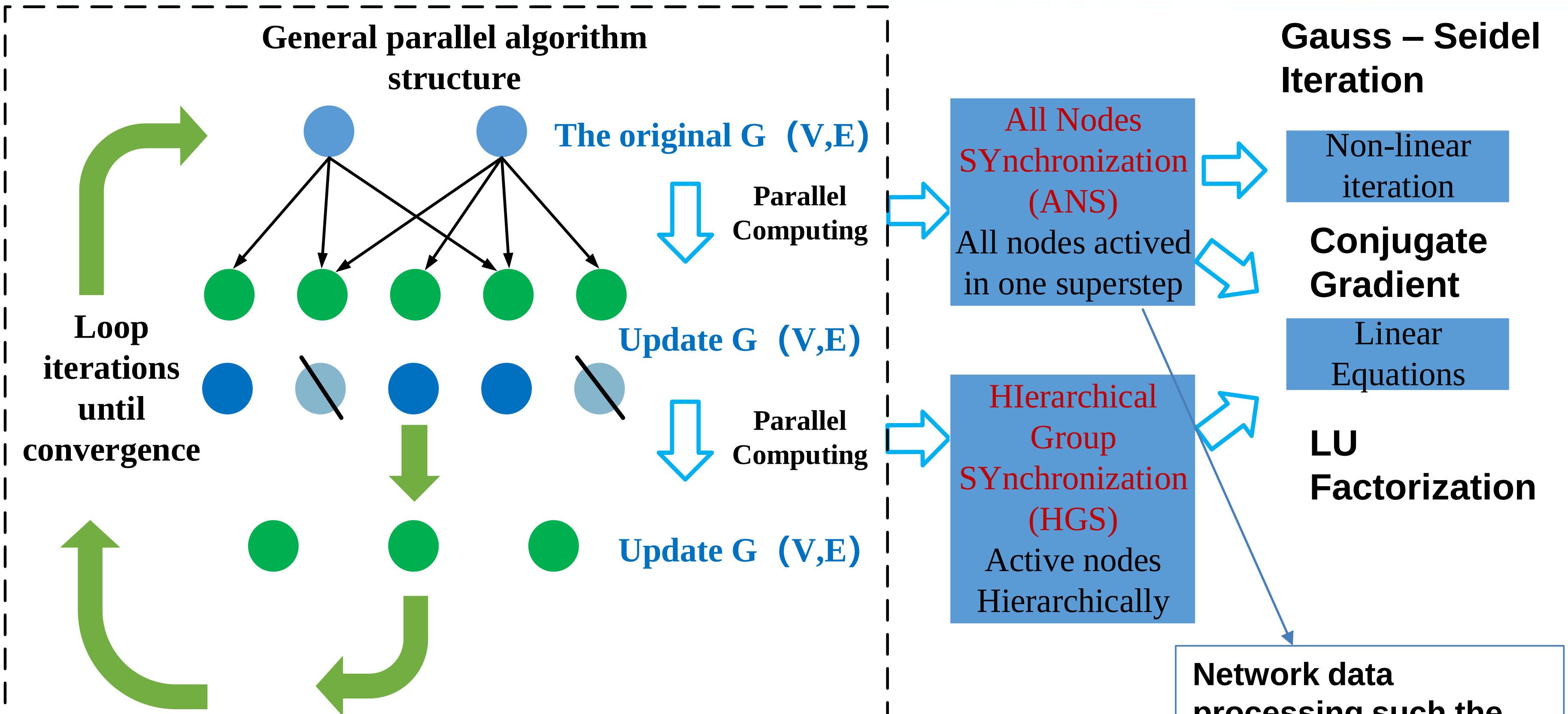


GDB Based Computing

Advantages of Graph Computing for Power System

- Integrate system modeling, core computing, and result visualization in graph database
- Data input preprocessing and result output traversal are not need
- Change calculation approach from serial computing to parallel queries
- Implement a suite of computation queries as library for power system applications

Graph Computation Modes



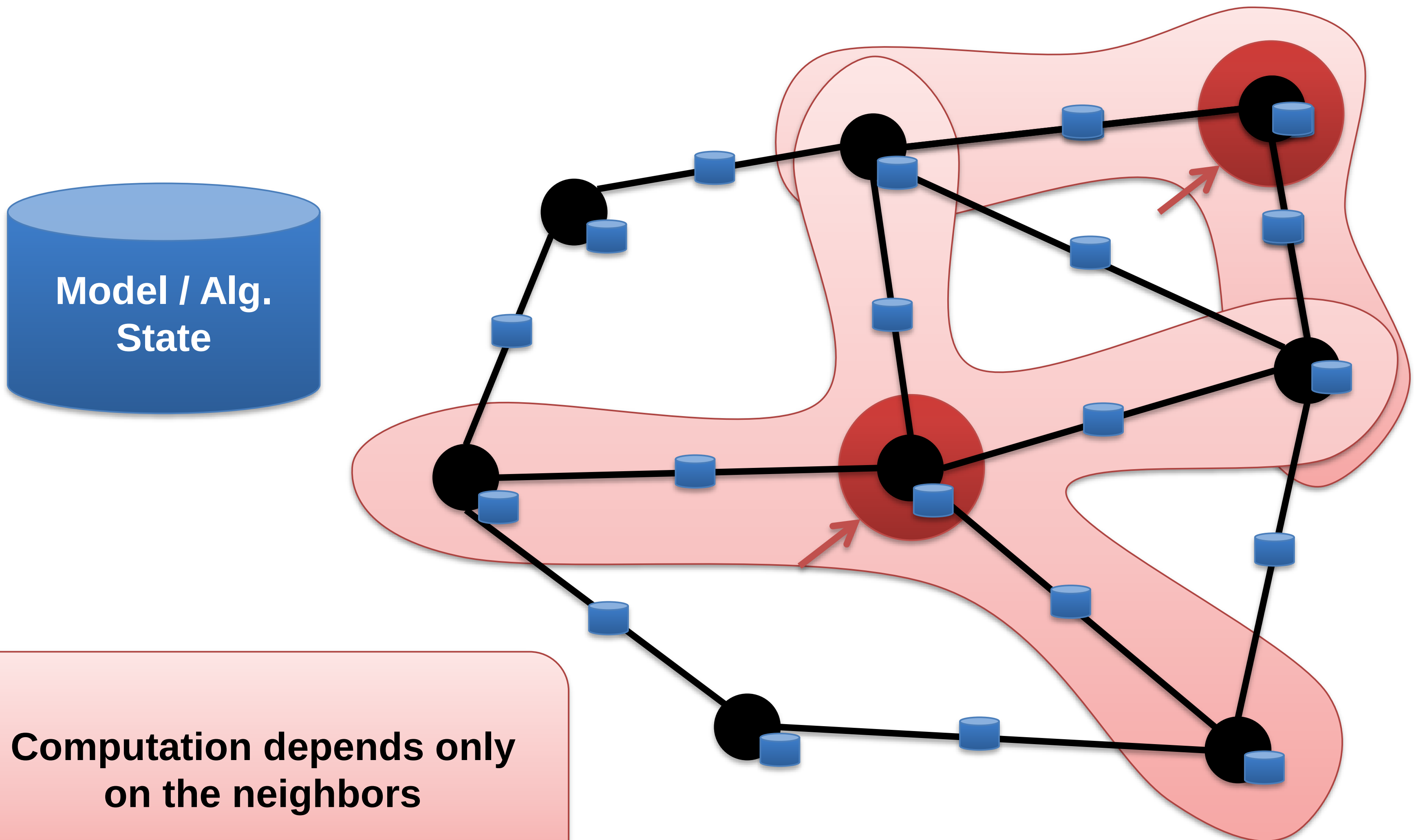
Network data processing such the formation of admittance, Jacobian and Gain matrix, the calculation of output, violation checking

- **Data-centric Parallel Mode**
  - Operations are defined on a group of vertices or edges = a frontier
  - Operations = manipulations of one or more frontiers

Leslie G. Valiant, A bridging model for parallel computation, Communications of the ACM, Aug. 1990
John Owens, Gunrock: A High-Performance, Data-Centric Abstraction for GPU Graph Computation, 2016

15

Vertex- oriented Graph-Parallel Computing

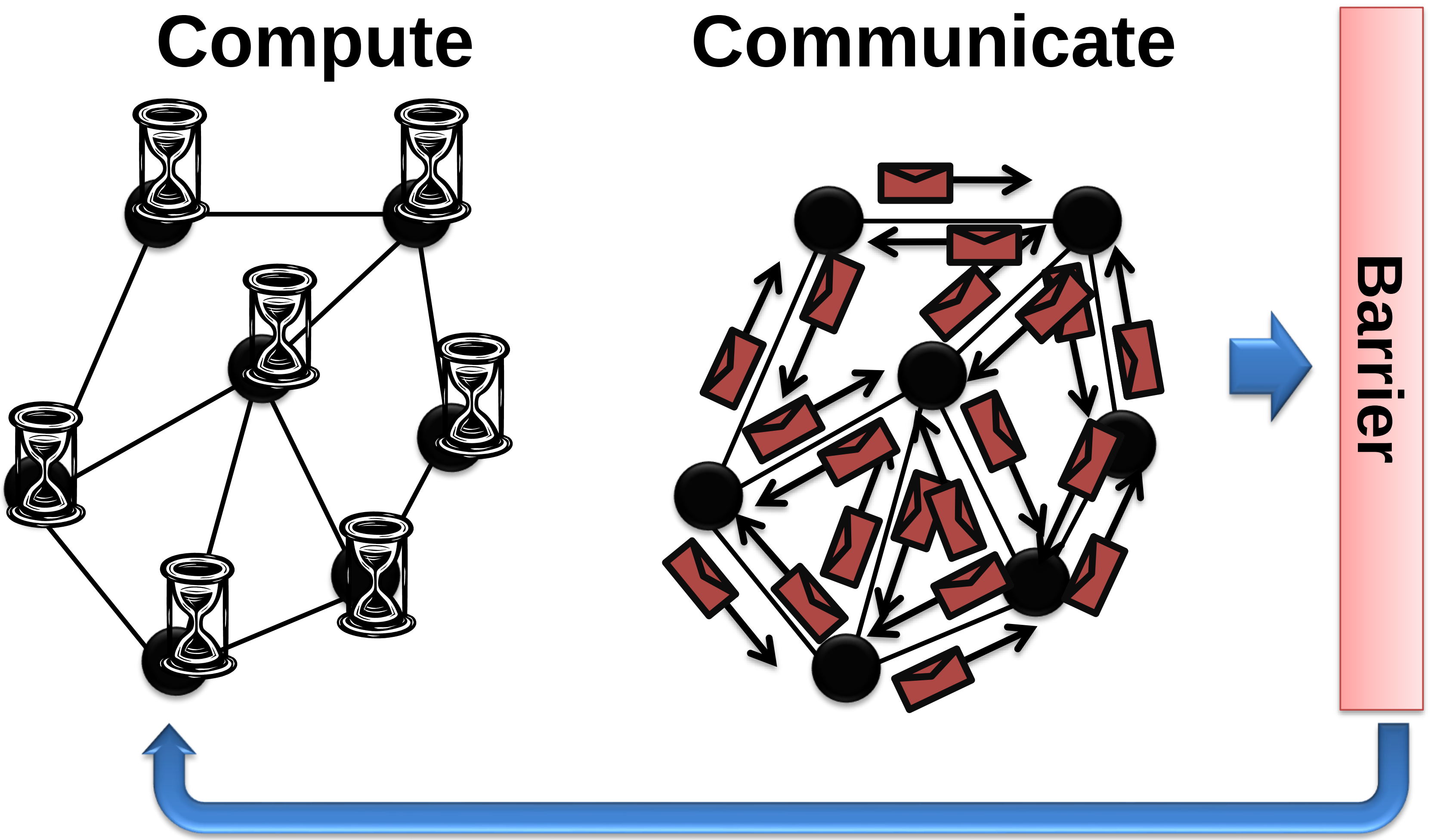


Computation depends only on the neighbors

Graph-Parallel Algorithms

- Collaborative Filtering
  - Alternating Least Squares
  - Stochastic Gradient Descent
  - Tensor Factorization
- Structured Prediction
  - Loopy Belief Propagation
  - Max-Product Linear Programs
  - Gibbs Sampling
- Semi-supervised ML
  - Graph SSL
- Community Detection
  - Triangle-Counting
  - K-core Decomposition
  - K-Truss
- Graph Analytics
  - PageRank
  - Personalized PageRank
  - Shortest Path
  - Graph Coloring
- Classification
  - Neural Networks

Pregel Bulk Synchronous Parallel Model

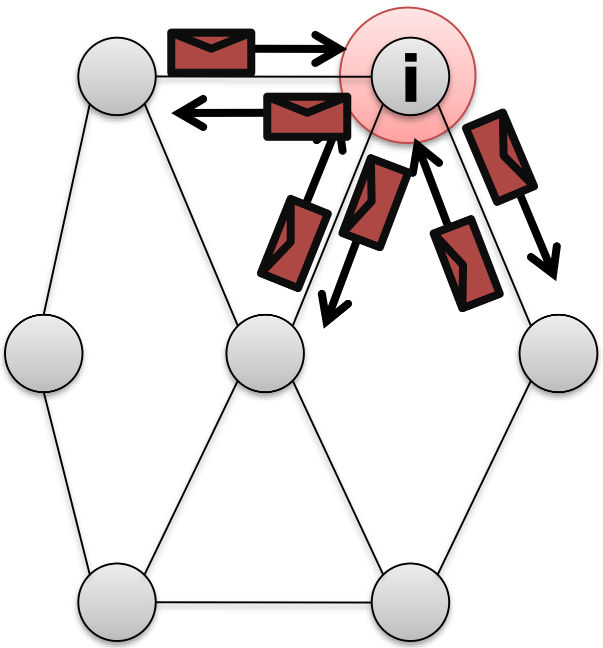


Vertex-Programs interact by sending messages.

PageRank based on Pregel

Widely used by google search engine

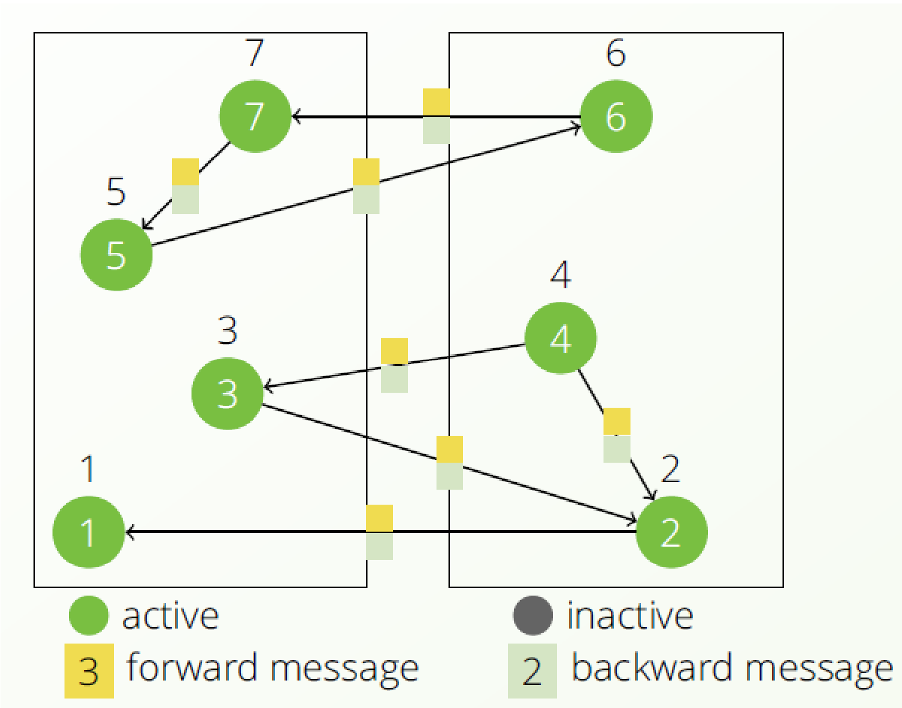
```
Pregel_PageRank(i, messages) :
  // Receive all the messages
  total = 0
  foreach( msg in messages) :
    total = total + msg

  // Update the rank of this vertex
  R[i] = 0.15 + total

  // Send new messages to neighbors
  foreach(j in out_neighbors[i]) :
    Send  msg(R[i]) to vertex j
```

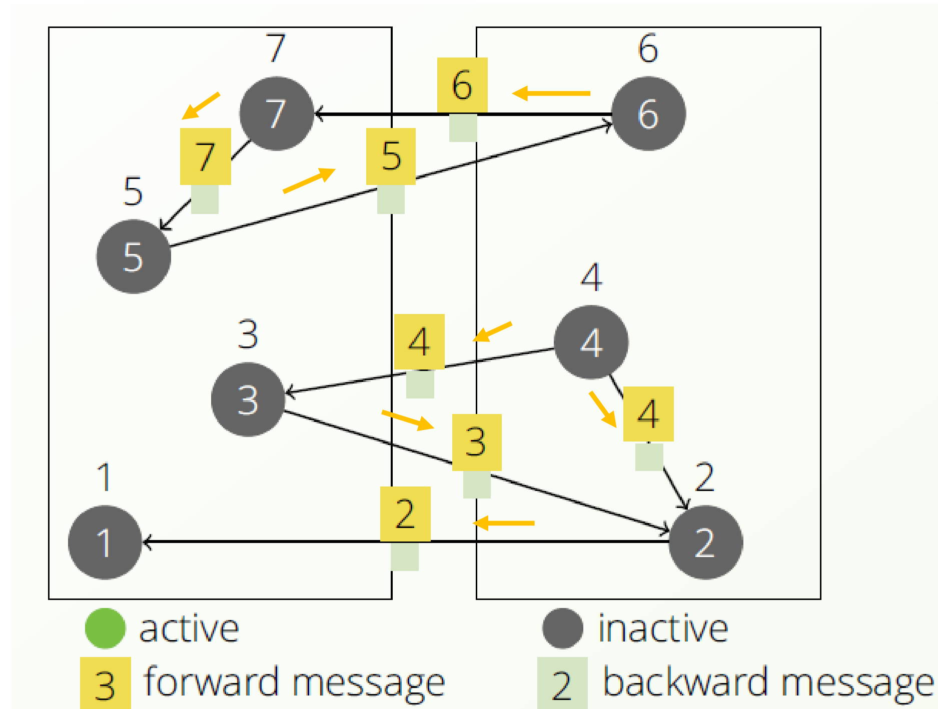


PageRank can be used to implement the parallel calculation of bus related in Power Flow, Sate Estimation

Malewicz et al. [PODC'09, SIGMOD'10]

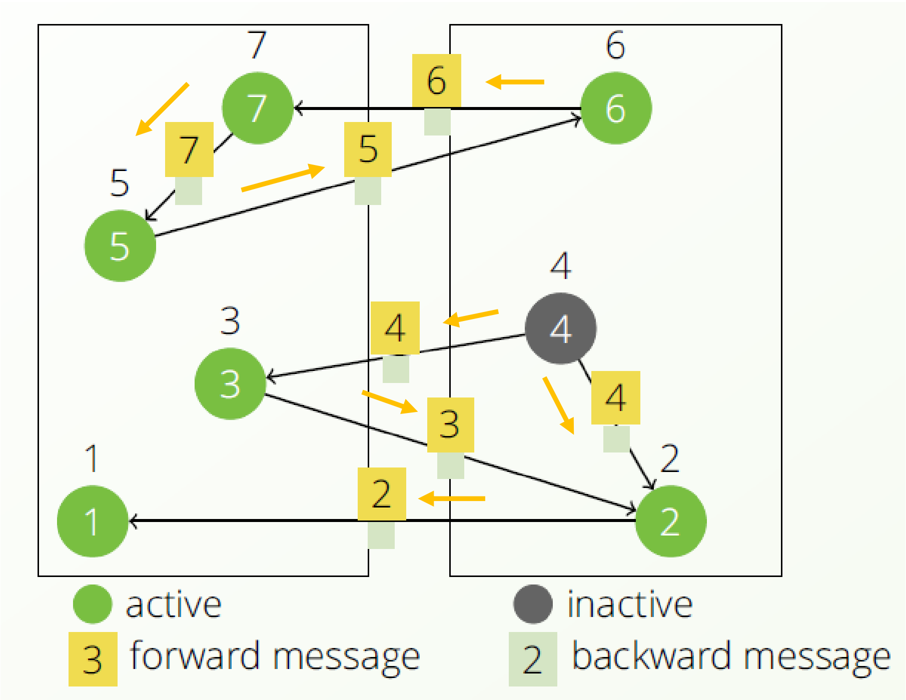
Connected Components based on Pregel



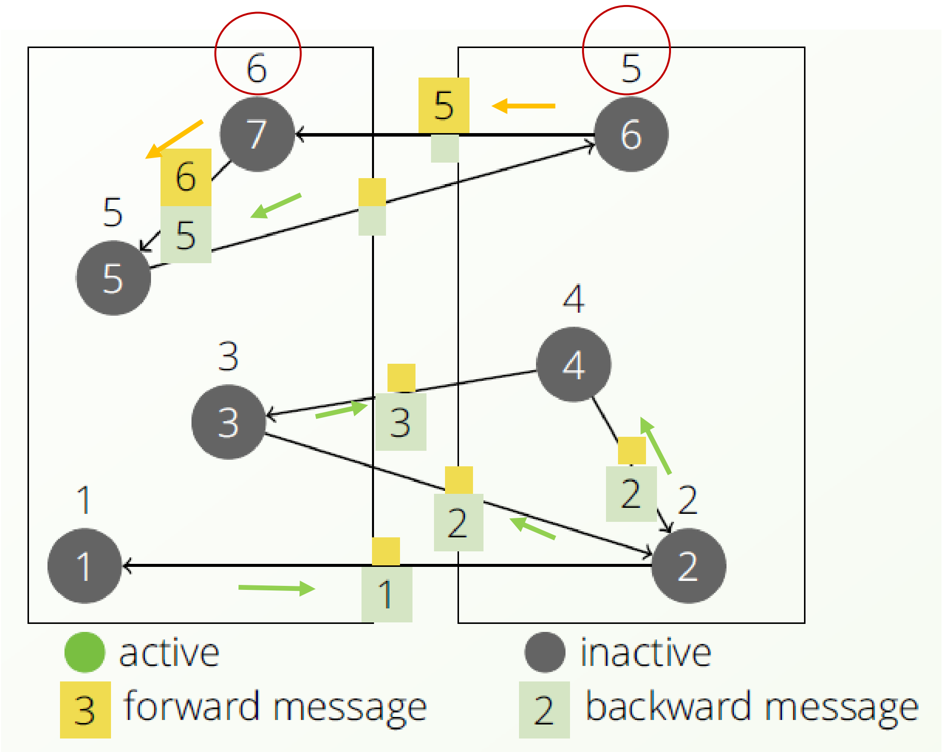


Send message

Connected Components based on Pregel

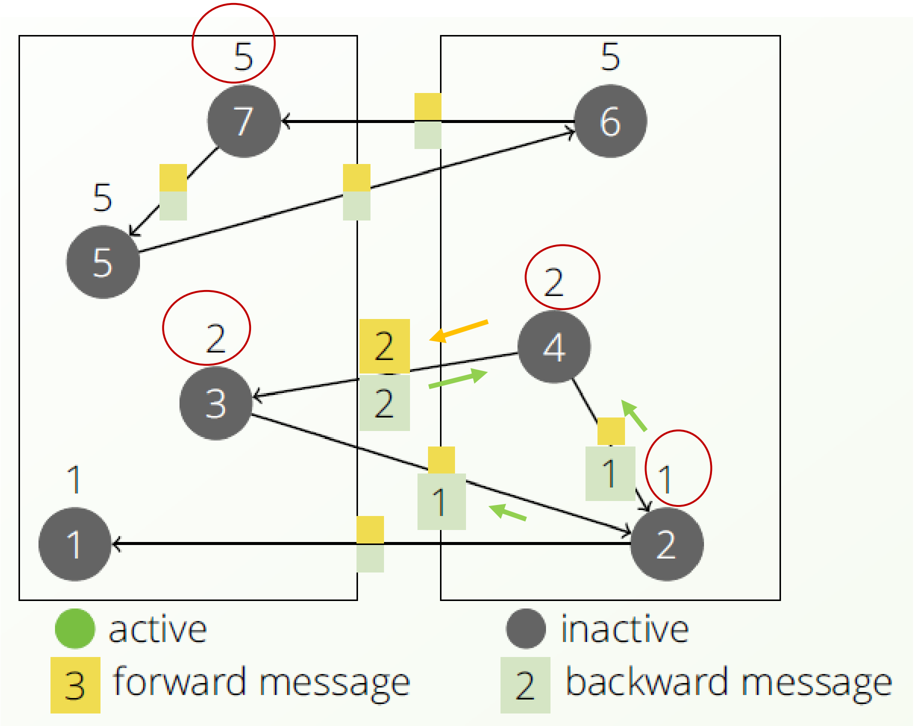


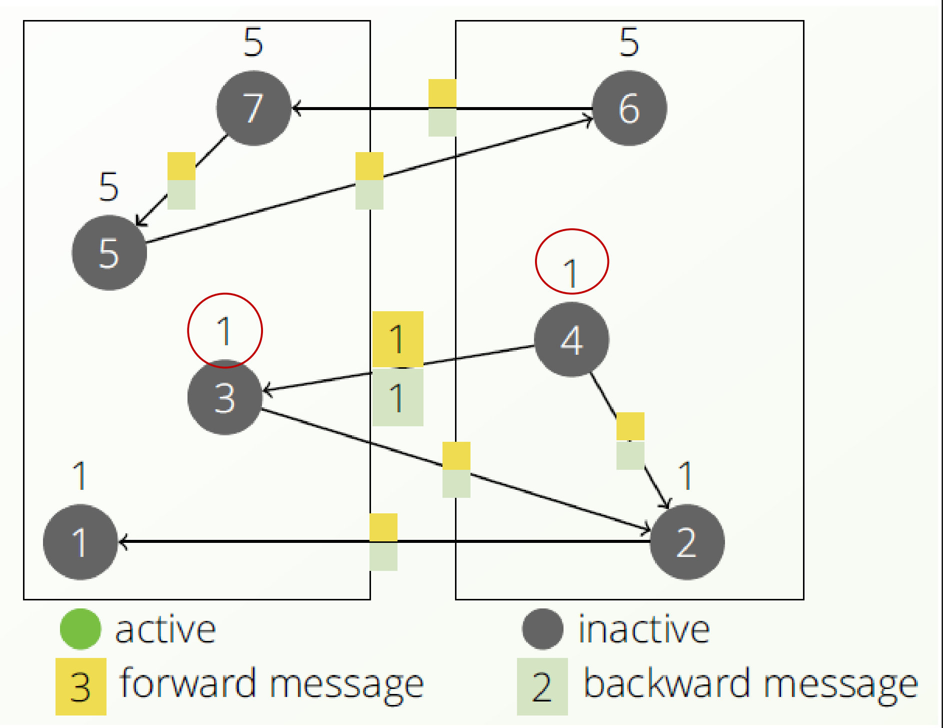
Send forward message



Send message and assign smaller value

Connected Components based on Pregel



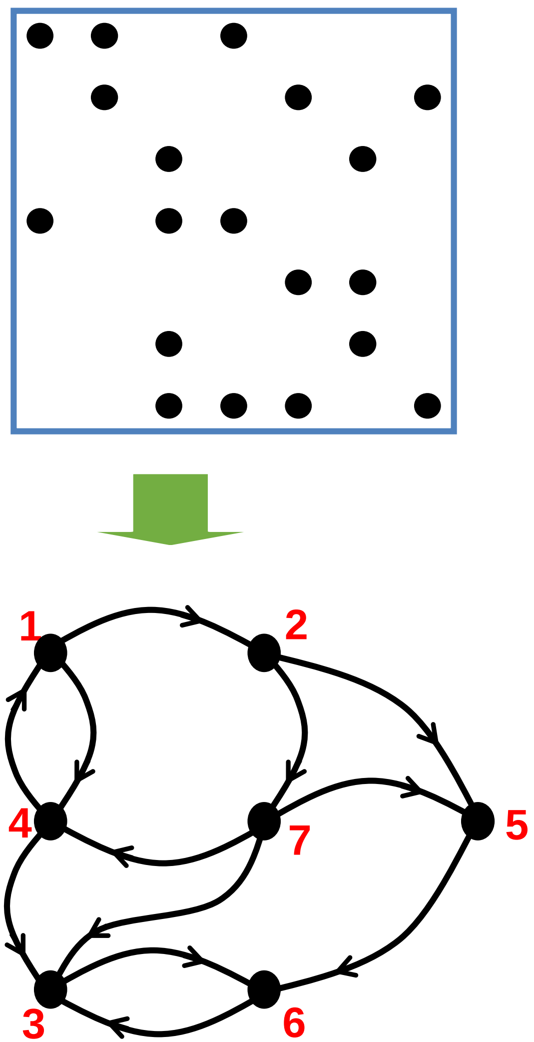


Send message and assign smaller value

Send message and assign smaller value

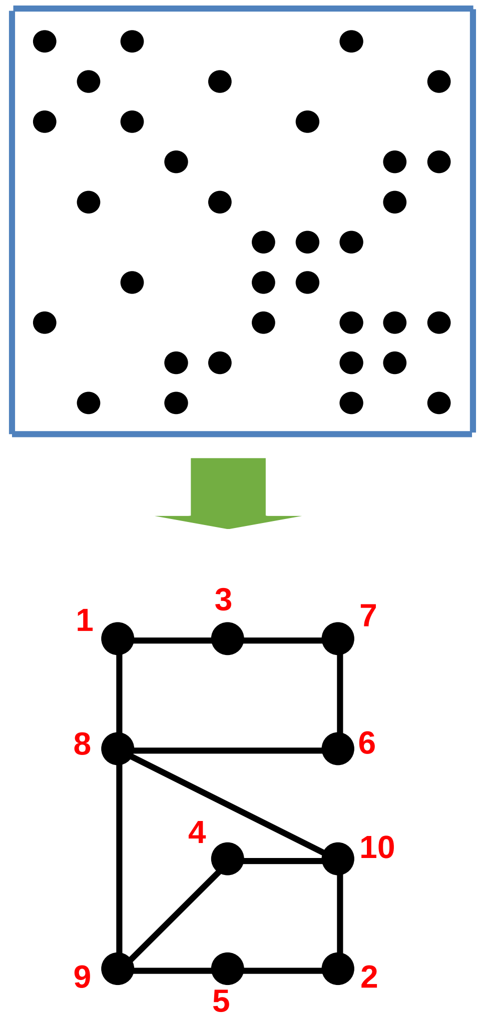
Matrix and Graph

Unsymmetrical Matrix



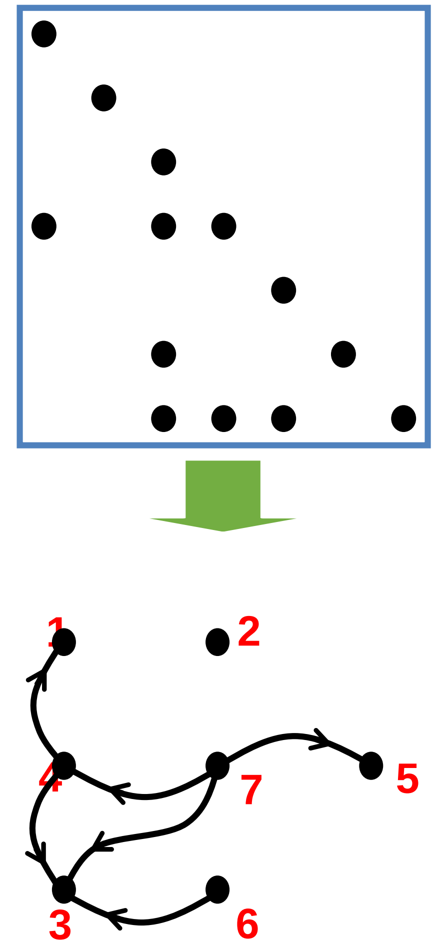
Directed Graph

Symmetric Matrix



Undirected Graph

Triangular Matrix



23 Directed Acyclic Graph

Graph LU and Solver

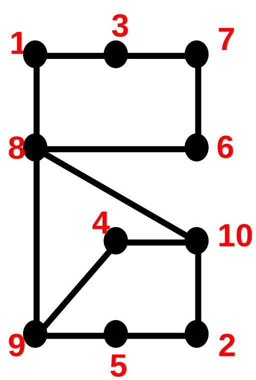
Graph Hierarchical Parallel Computing

1.Min-degree Order	2. E-tree Symbolic factorization	3.Numeric factorization	4.Triangular Solves
Graph Structure	Graph Modification	Graph Prosperities	Graph Prosperities

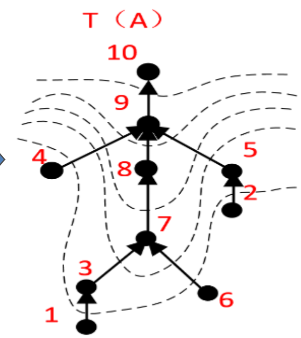
Three Types of Graph Computation:

(1) Graph structure;
(2) Graph addition and deletion;
(3) Graph properties calculation

Node Parallel

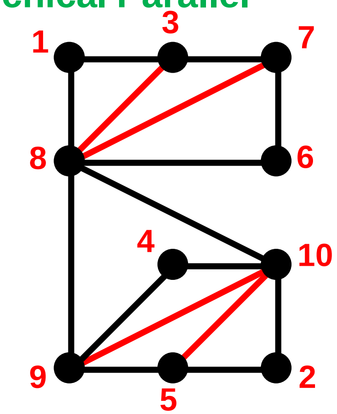


G(A)



T(A)

Hierarchical Parallel



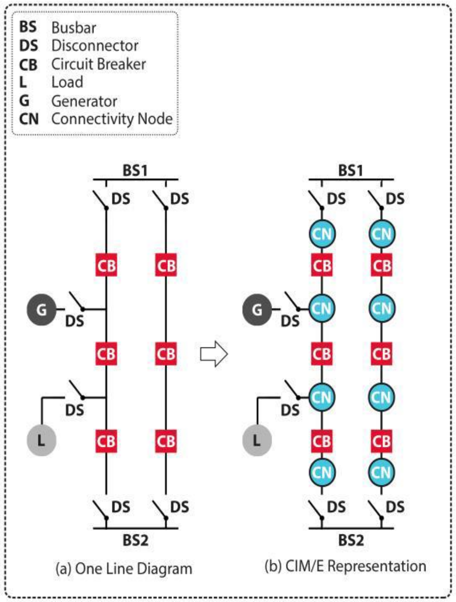
$G^+(A)$

Level	Nodes
1	1, 2, 4, 6
2	3,5
3	7
4	8
5	9
6	10

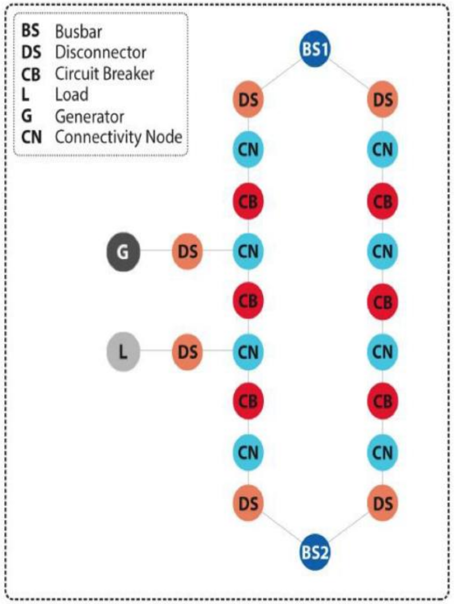
Node Hierarchical

Graph Structure Computation： Non-zero counts, nonzero structure of A, Elimination Tree；
Graph Modification Computation： Create/modify T(A) and G+(A)， numeric factorization；
Graph Properties Calculation： Triangular solves on G+(A)

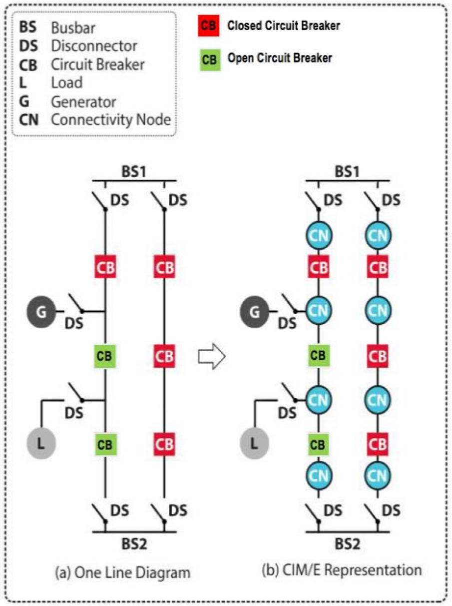
GDB for Node-Breaker Model



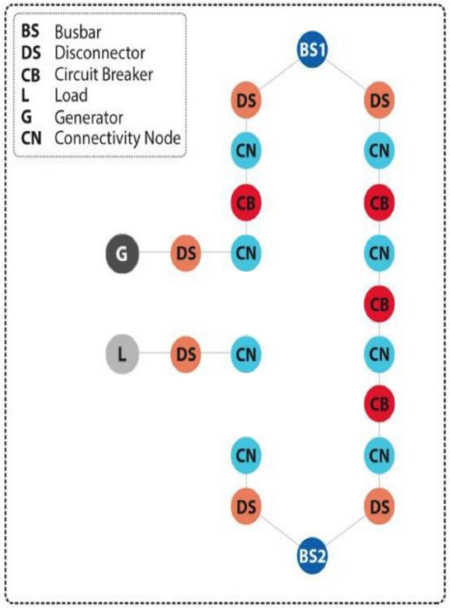
Substation representation in one-line diagram and CIM/E



Substation modeling in CIMGDB



Substation representation in one-line diagram and CIM/E



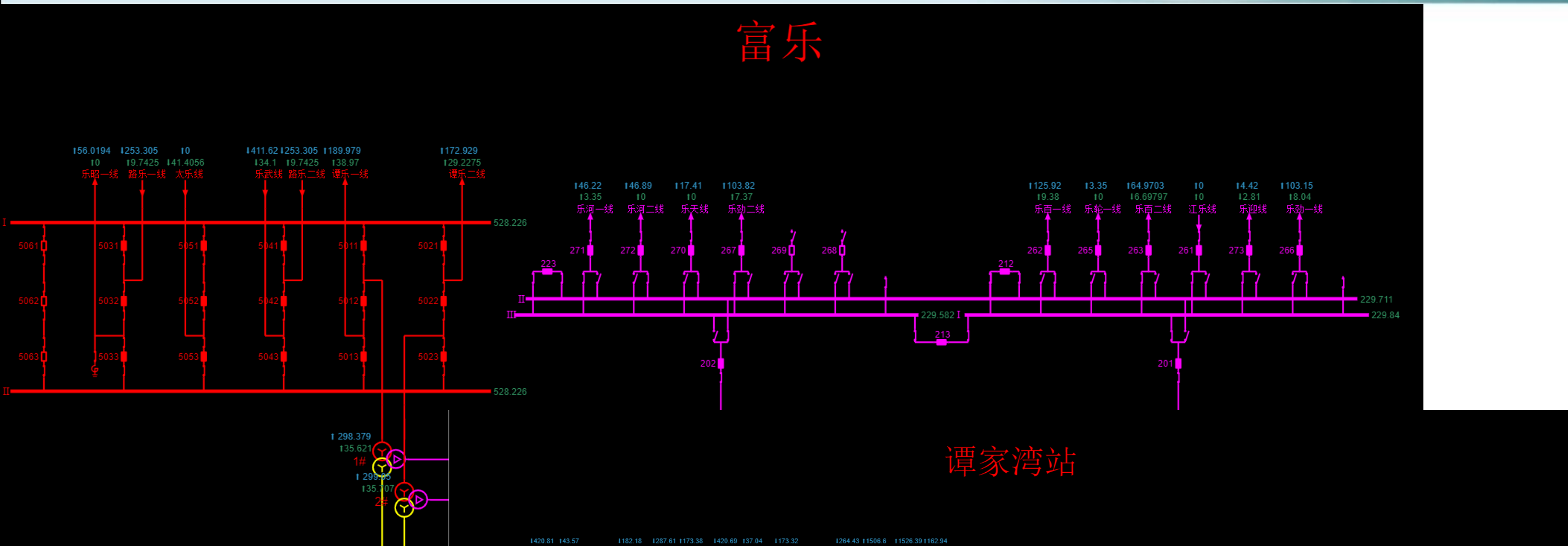
Substation modeling in CIMGDB

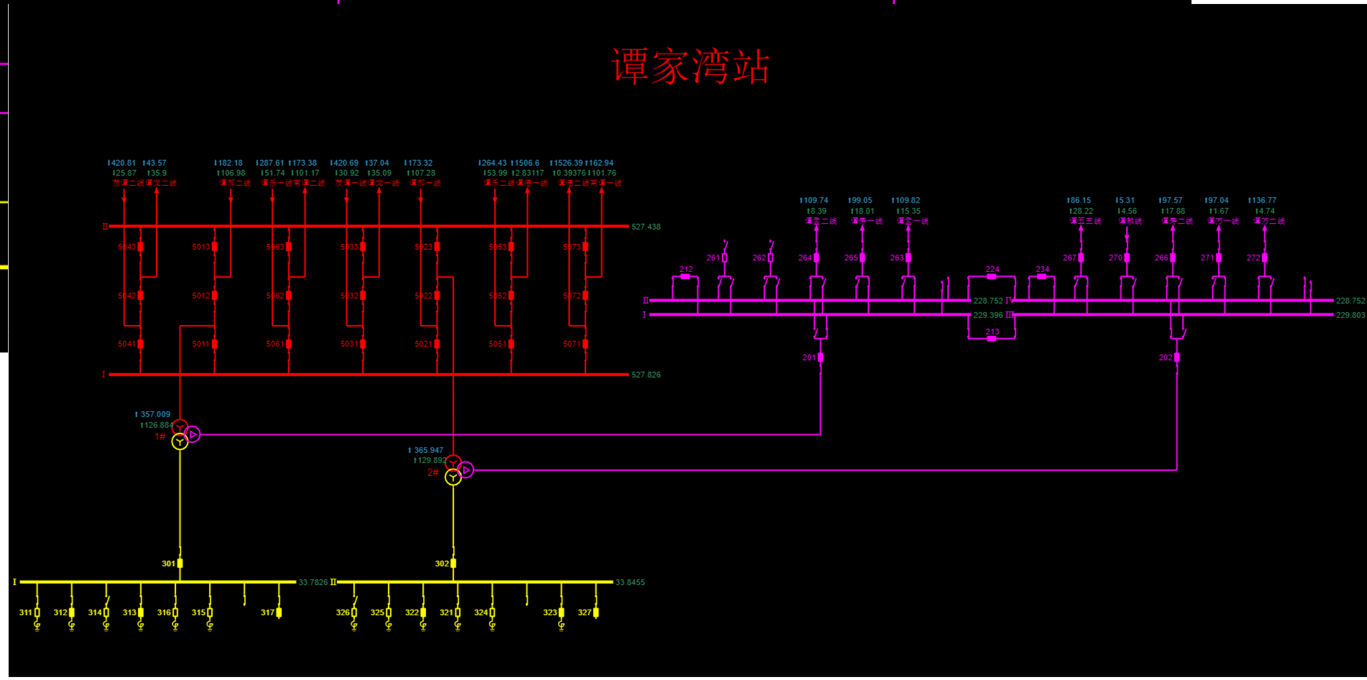
Node – Breaker Graph Model

- ❑ Based on Base-value, Substation, Bus, AC line, Unit, Transformer, Load, Compensator, Converter, DC line, Island, Topo-node, Breaker and Disconnector are modeled by vertices
- ❑ Common Information Model (CIM)

Connected Components is used to convert Node-Breaker to Bus-Branch Graph

Substation One-line Diagram





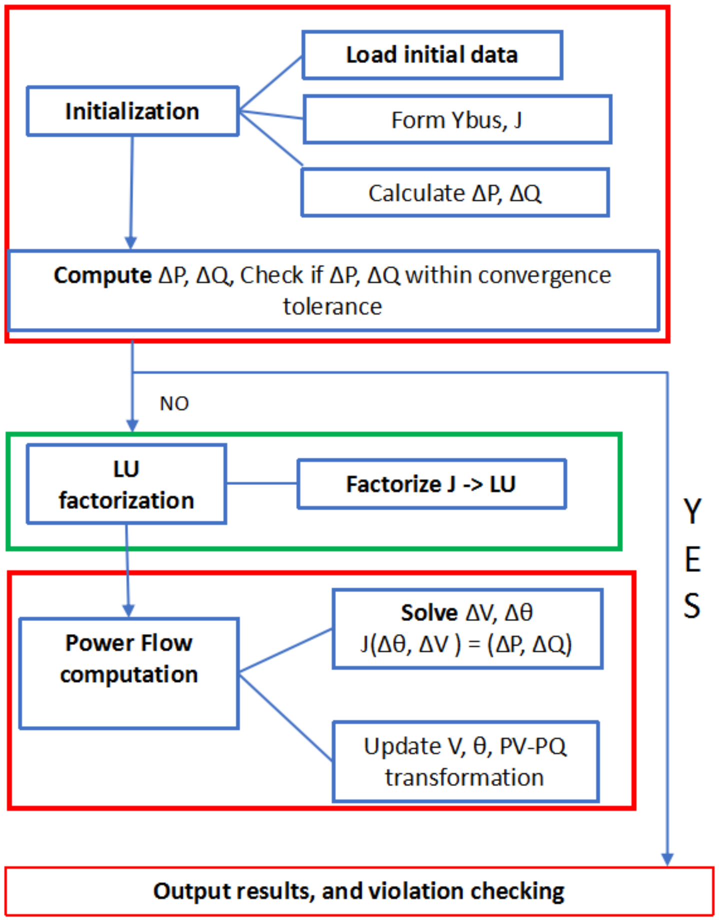
Substation one-line diagram created automatically by Graph Analysis

Graph Based N-R and F-D Power Flow Algorithms

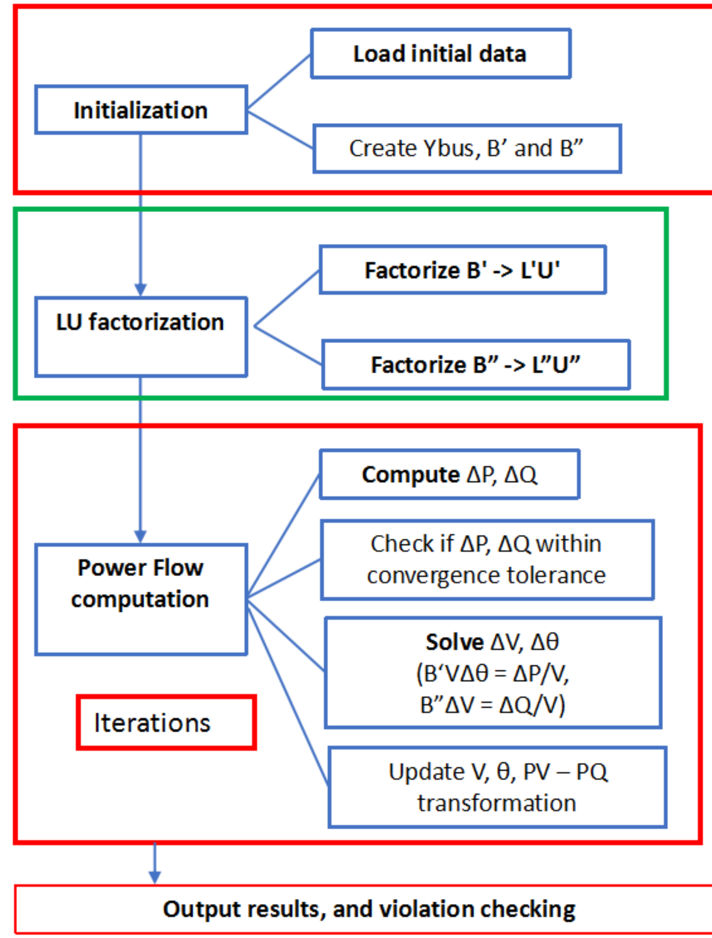
STATE GRID
GEIRI NORTH AMERICA
全球能源互联网美国研究院

Page Rank Node parallel Computation

GraphLU Solver Hierarchical Parallel Computation



Newton-Raphson



Fast Decoupled

Graph Based State Estimation Algorithm

STATE GRID
GEIRI NORTH AMERICA
全球能源互联网美国研究院

$$\begin{cases} G_{AA}\cdot\Delta\theta = {H_{AA}}^T R_A^{-1}(z_A - h_A(x)) \\ G_{RR}\cdot\Delta|V| = {H_{RR}}^T R_R^{-1}(z_R - h_R(x)) \end{cases}$$

$$\begin{cases} G_{AA} = {H_{AA}}^T R_A^{-1} H_{AA} = \sum_i^{n-1} {H_{AA,i}}^T \cdot {R_{A,i}}^{-1} \cdot H_{AA,i} = \sum_{i=1}^{n-1} G_{AA,i} \\ G_{RR} = {H_{RR}}^T R_R^{-1} H_{RR} = \sum_i^{n} {H_{RR,i}}^T \cdot {R_{R,i}}^{-1} \cdot H_{RR,i} = \sum_{i=1}^{n} G_{RR,i} \end{cases}$$

Page-Rank Node parallel Computation

GraphLU Solver Hierarchical Parallel Computation

1. Start Iterations, set iteration index $k = 0$;
2. Initialize the system state vector x^k, including θ^k and $|V|^k$ (flat start or not);
3. Formulate gain matrices, G_{AA} and G_{RR}, based on Page-Rank node parallel computing;
4. Decompose G_{AA} and G_{RR} using parallel LU solver;
5. Update right-hand-side vector ${H_{AA}}^T R_A^{-1}\left(z_A - h_A(x^k)\right)$ based on Page-Rank node parallel computing, solve $\Delta\theta^k$, and update $\theta^{k+1} = \theta^k + \Delta\theta^k$;
6. Check convergence: $\max|\Delta x^k| \le \epsilon$? If yes, output θ^{k+1} and $|V|^k$; If no, go to step 7;
7. Update right-hand-side vector ${H_{RR}}^T R_R^{-1}\left(z_R - h_R(x^k)\right)$ based on Page-Rank node parallel computing, solve $\Delta|V|^k$, and update $|V|^{k+1} = |V|^k + \Delta|V|^k$;
8. Check convergence: $\max|\Delta x^k| \le \epsilon$? If yes, output θ^{k+1} and $|V|^{k+1}$; If no, $k = k + 1$, go to step 5;

Graph Computing: A Real Case

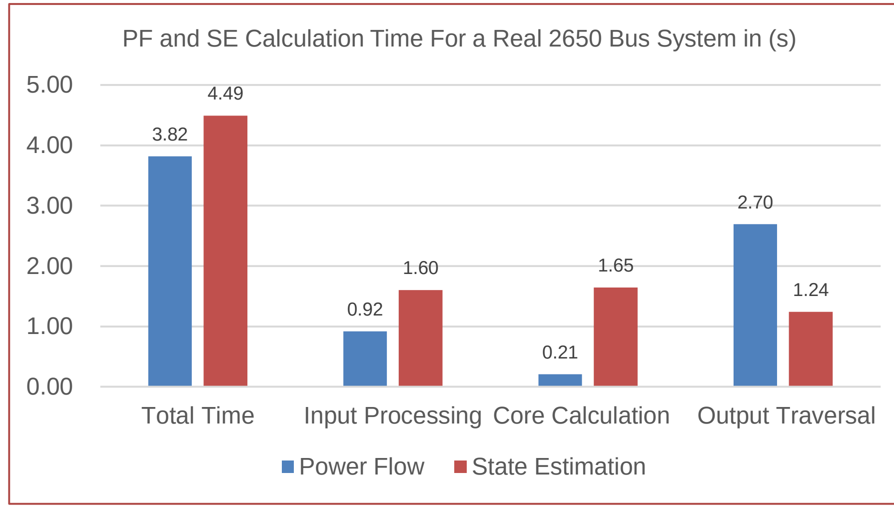
STATE GRID
GEIRI NORTH AMERICA
全球能源互联网美国研究院

Relational Database

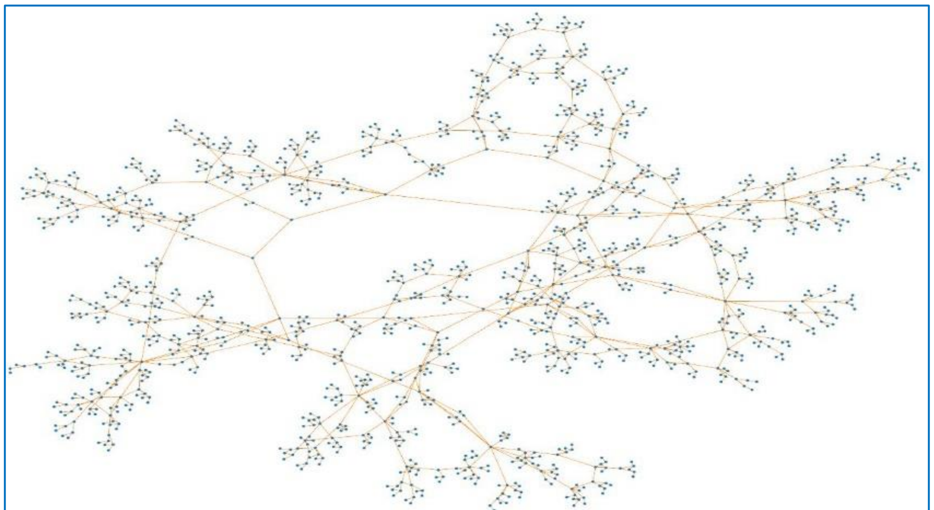
- <u>Preprocessing:</u> Search bus tables and branch tables to find connectivities.
- <u>Output Traversal:</u>Map solved variables to bus voltage and branch flow.

Graph Database

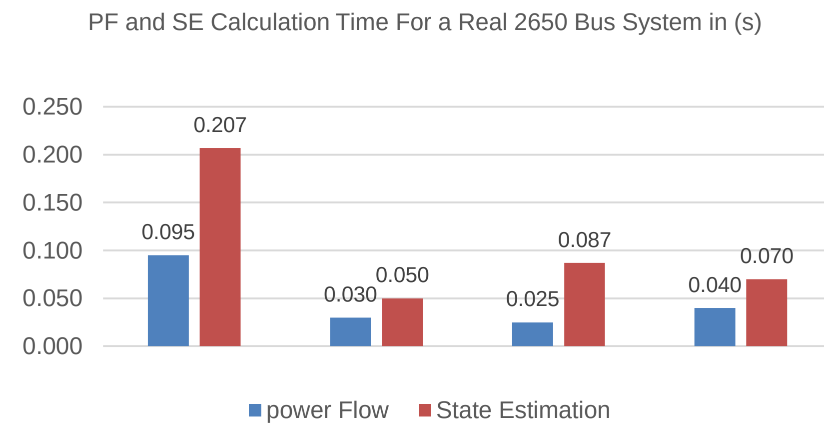
- **No preprocessing**. Connectivity is predefined in graph database.
- **No output traversal**. Solved bus voltage and branch flow are attributes of nodes and edges in graph database.



Relational Database

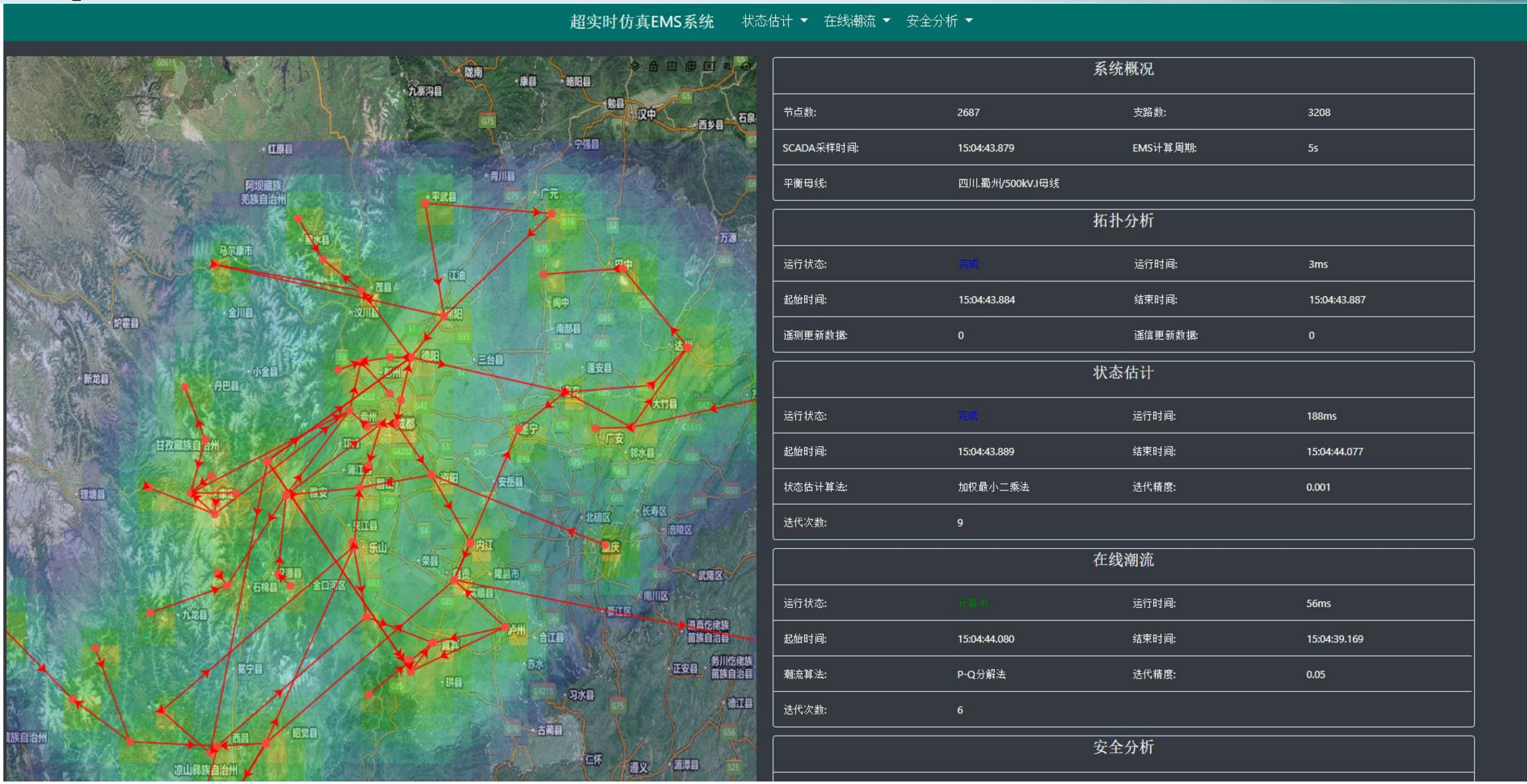


2650 Bus System Model in GDB



Graph Database and Computing

Graph Based
Super-fast EMS Platform



- ❑ SE/PF/CA including topology processing is faster than real time, completed within 5 seconds
- ❑ Visualize applicatoin status, start time, and execution time
- ❑ Voltage heat map and operational serverity index show overall operational risk
- ❑ Substation diagram is automatically dynamically drawn and shows the detailed operations

Conclusions

STATE GRID
GEIRI NORTH AMERICA
全球能源互联网美国研究院

- EMS cycle shall be speeded up as power system is significantly evolving to be larger and more complex with more power electronics, higher uncertainty, and faster events.
- Fast and predictive analytics are critical to respond the cascading events, avoid the severe blackouts, and enable the advanced system automatic control.
- High performance computing is critical to meet the gap on the pathway to EMS Robot.
- Graph database and graph parallel computing are promising to achieve Super-fast EMS.

Thank You!

Q&A