Distributed Asynchronous Algorithms and Software Systems for Wide-Area Monitoring of Power Systems

Aranya Chakrabortty[*], Frank Mueller[*], Rakesh Bobba[+], Nitin Vaidya[+] and Yufeng Xin[++]

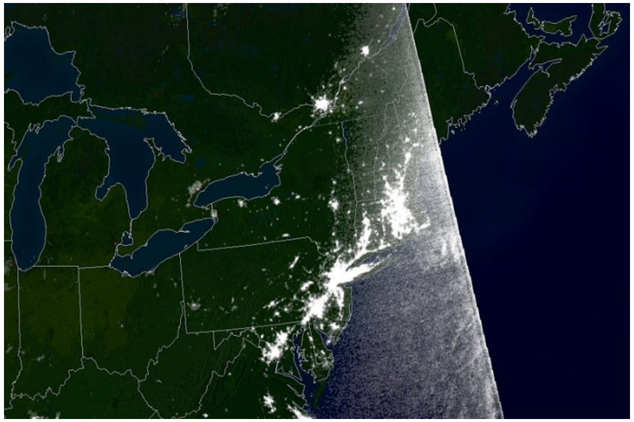
[*] North Carolina State University, [+]University of Illinois Urbana Champaign,
[++]RENCI, University of North Carolina

NSF CPS PI Meeting

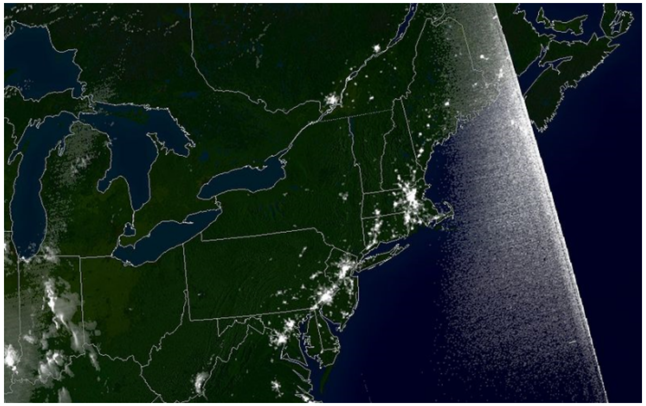
November 16, 2015, Arlington, VA

Main trigger: 2003 Northeast Blackout

NYC before blackout



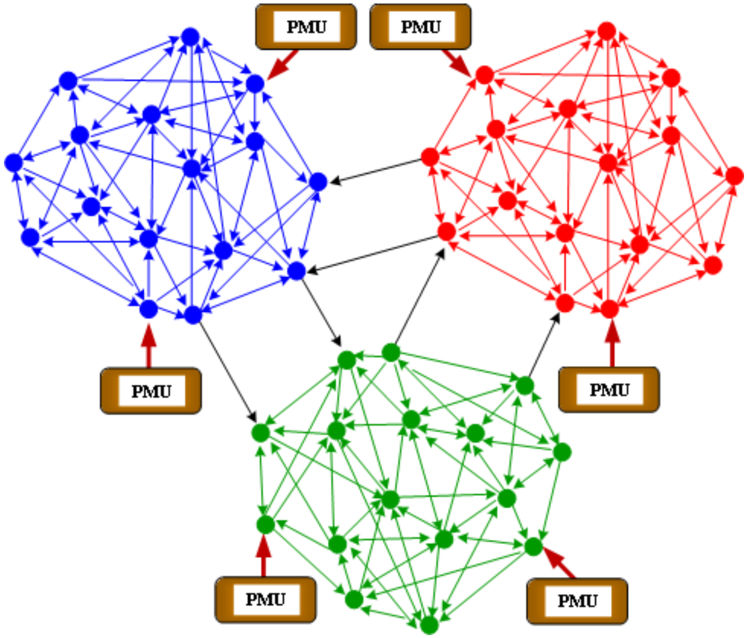
NYC after blackout



2 Main Lessons Learnt from the 2003 Blackout:

1. Need significantly higher resolution measurements

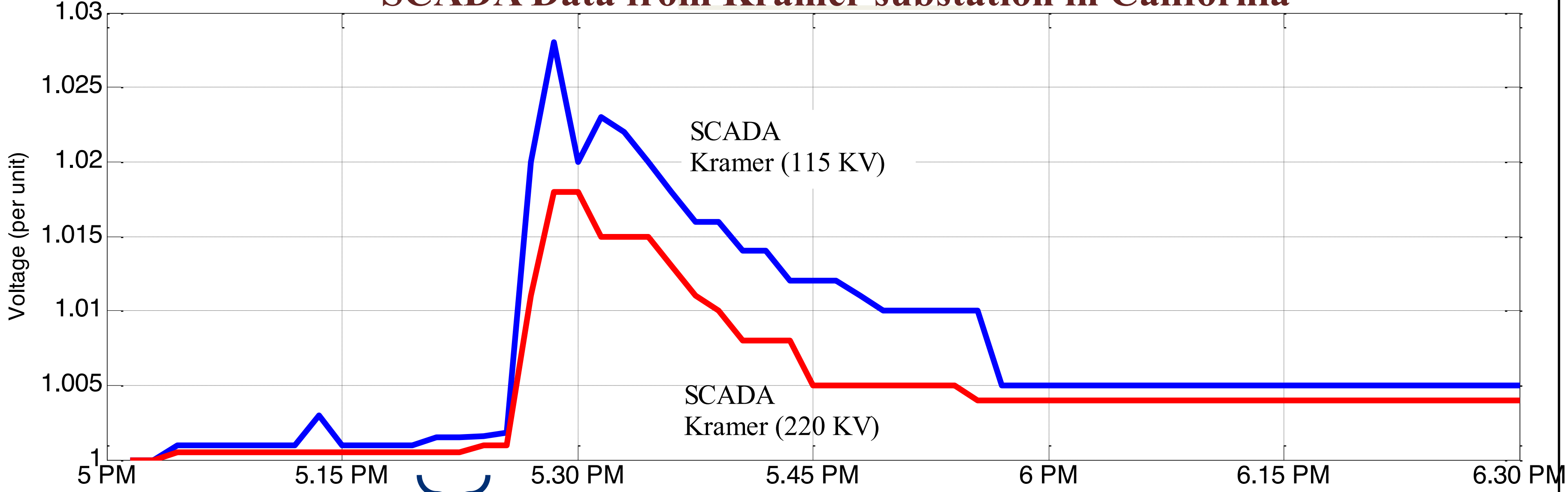
⇨ From traditional SCADA (System Control and Data Acquisition) to PMUs (Phasor Measurement Units)



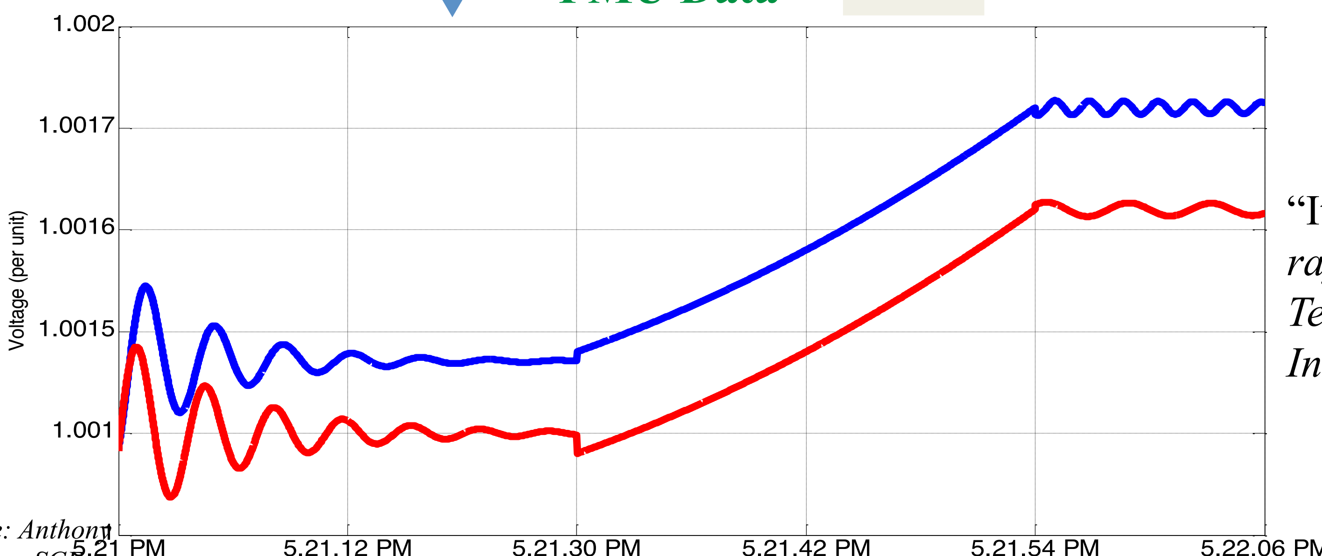
2. Local monitoring & control can lead to disastrous results

Hauer, Zhou & Trudnowsky, 2004
Kosterev & Martins, 2004

SCADA Data from Kramer substation in California



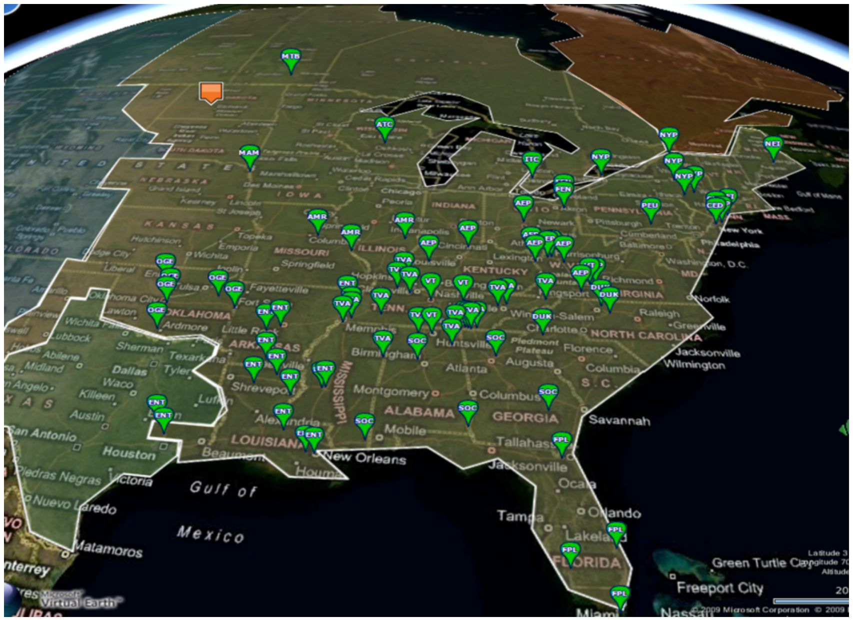
PMU Data

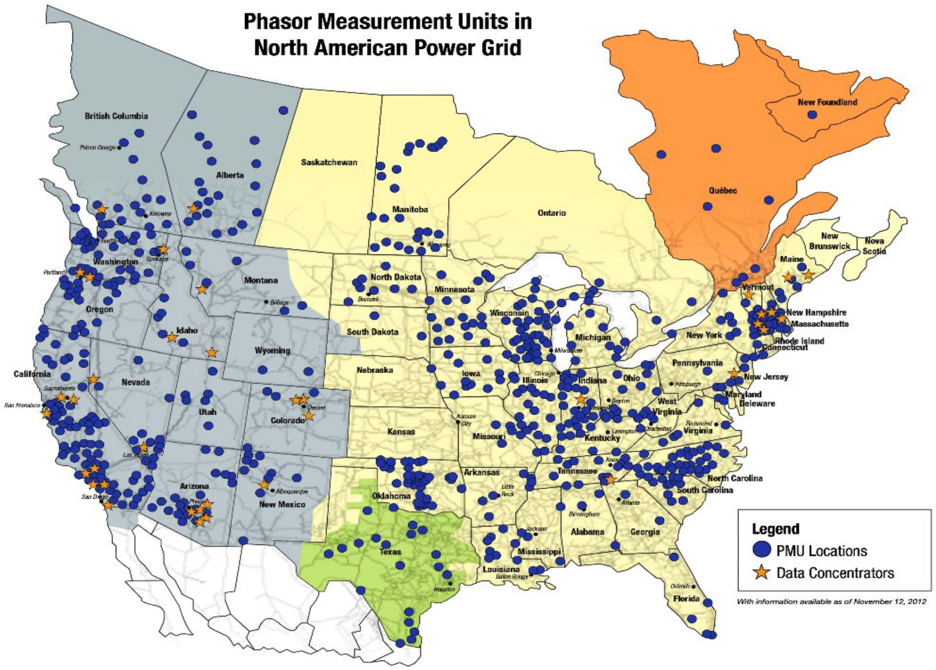


“It's like going from an *X-ray* to a *MRI* of the grid.”
Terry Boston, CEO, PJM Interconnection

Source: Anthony Johnson, SCE

Increasing Volumes of PMU Data



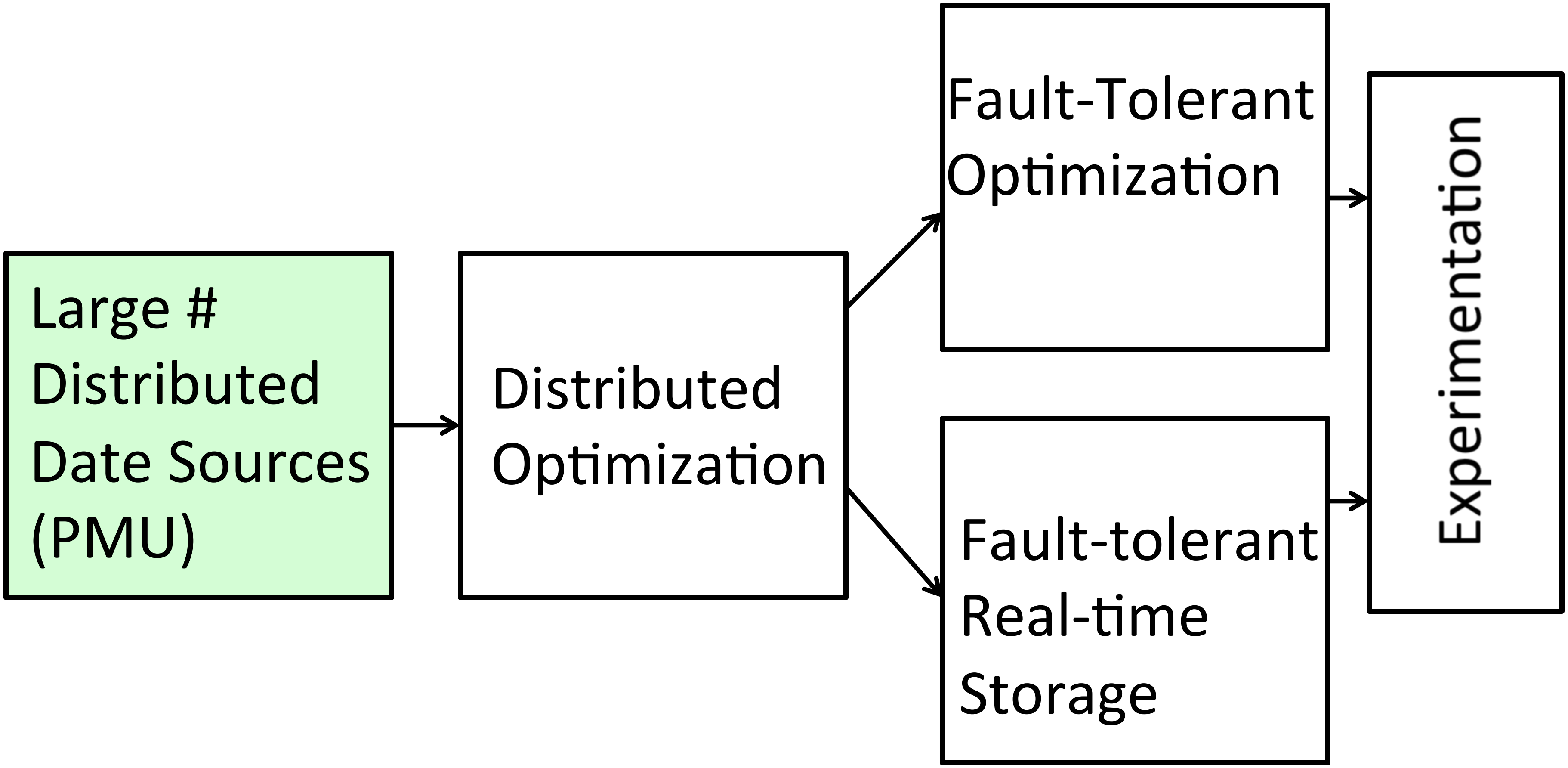


2008: Only 40 PMUs in the entire east coast

2015: More than 1200 PMUs across USA
(Nearly 52 PMUs only in North Carolina)

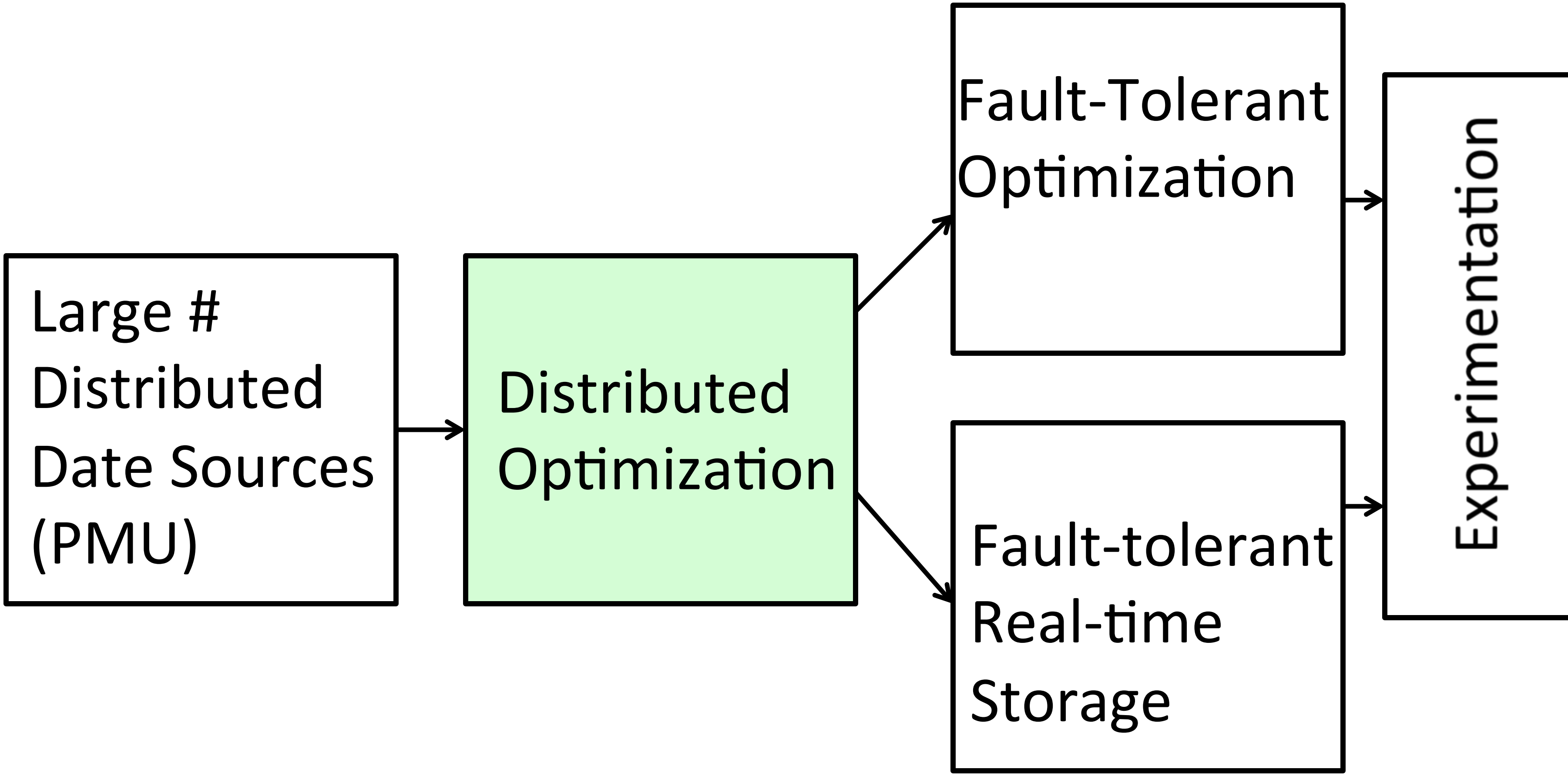
- Massive volumes of PMU
- Centralized processing will not be tenable

Outline



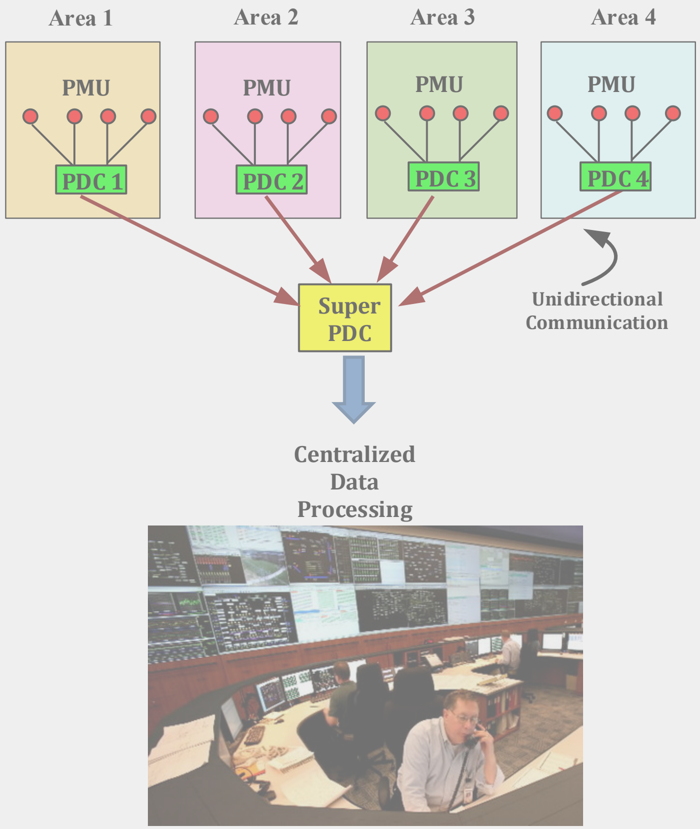
Large # Distributed Date Sources (PMU)
Distributed Optimization
Fault-Tolerant Optimization
Fault-tolerant Real-time Storage
Experimentation

Outline


Large #
Distributed
Date Sources
(PMU)
Distributed
Optimization
Fault-Tolerant
Optimization
Fault-tolerant
Real-time
Storage
Experimentation

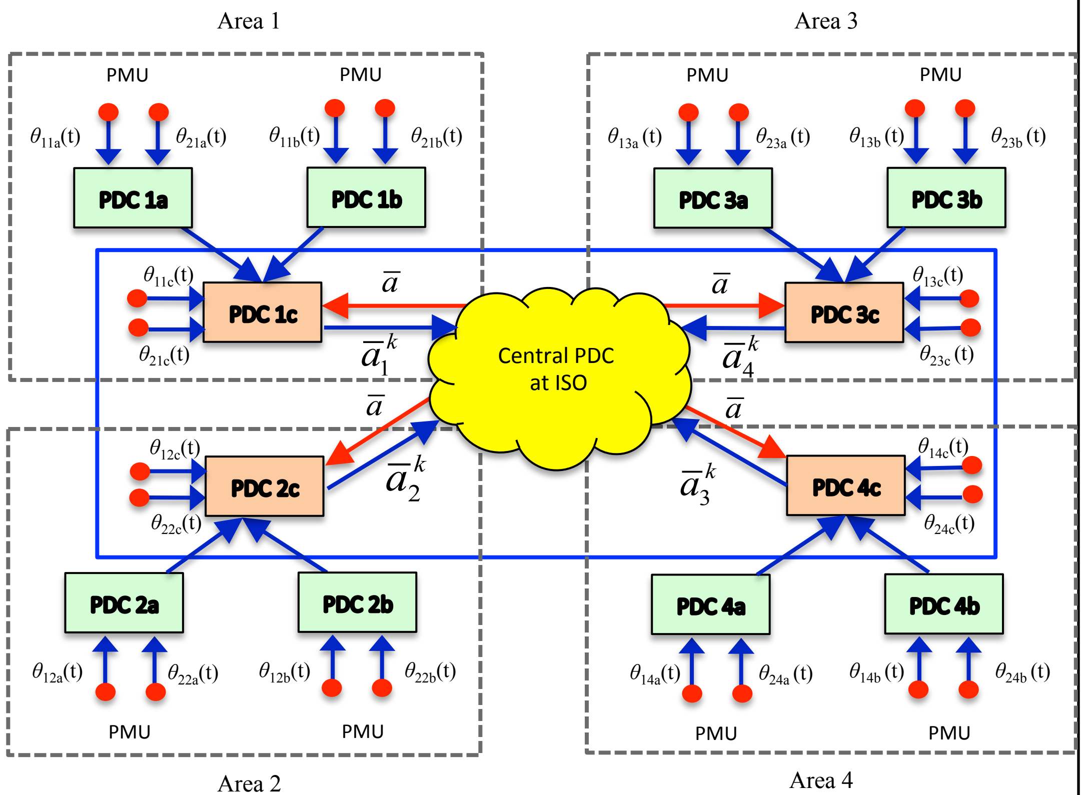
Centralized vs Distributed Algorithms

Centralized WAMS



Control Room

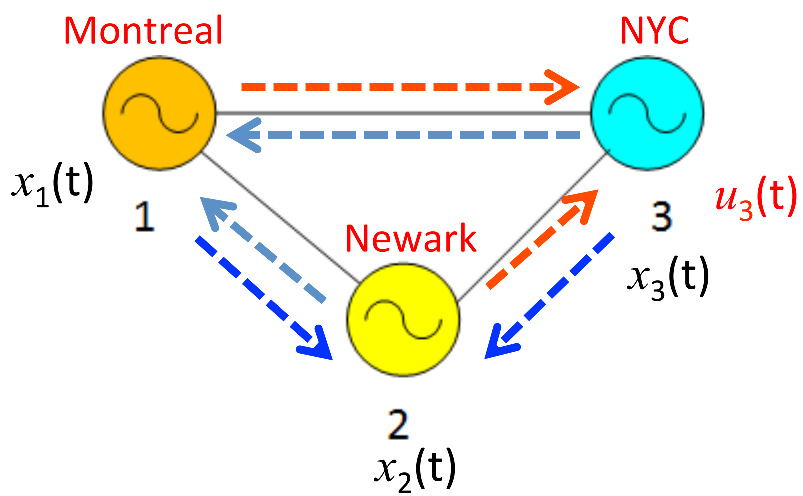
(Hierarchically) Distributed



Specific application of interest for this talk:

Wide-area oscillation monitoring

Distributed Computation



Heavy online computation with volumes of data transfer in unsecured network connecting generation sites directly

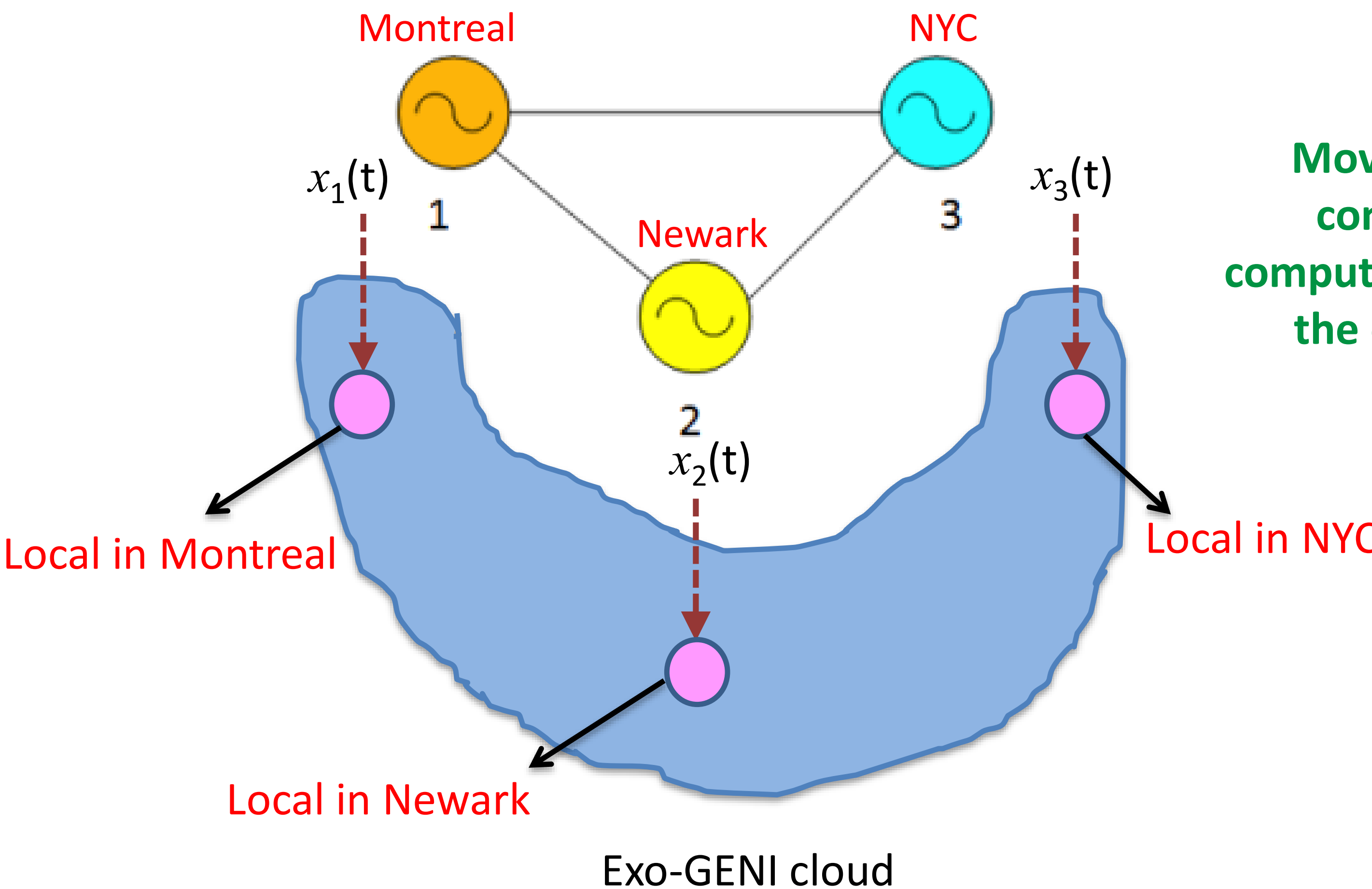
Swing equation model:

$$\begin{bmatrix} \Delta\dot{\delta} \\ M\Delta\dot{\omega} \\ \Delta\dot{E} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ -L(G) & -D & -P \\ K & 0 & J \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta\omega \\ \Delta E \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \mathrm{col}_{i=1\,(1)n}(\gamma_i) \\ \mathrm{col}_{i=1\,(1)n}(\rho_i) \end{bmatrix}}_{\text{due to load}} + \begin{bmatrix} 0 & 0 \\ 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} \Delta P_m \\ \Delta E_F \end{bmatrix}$$

L(G) = fully connected network graph

Controllable inputs

Distributed Computation in the Cloud

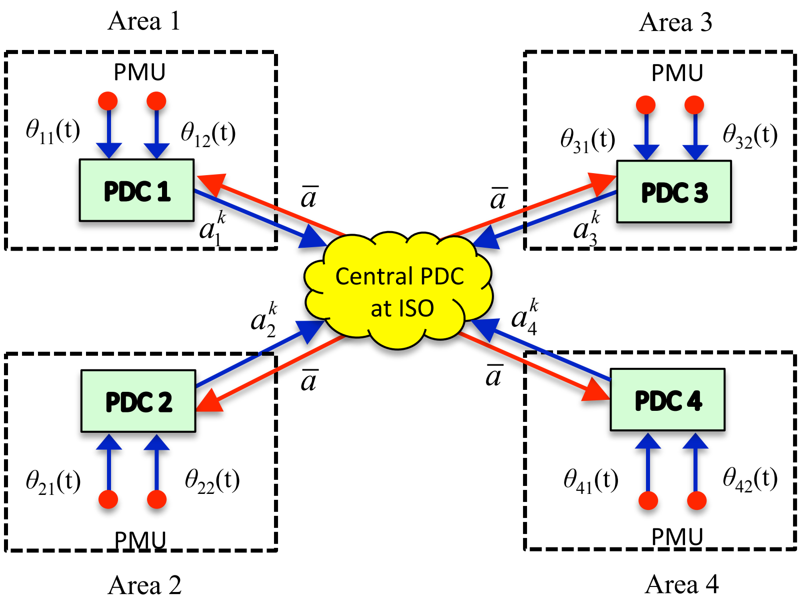


Move the control computations to the cloud

Computation can happen in local clouds with cloud-to-cloud communication instead of gen-to-gen communication

Wide-Area Oscillation Estimation

Distributed:



Multiple Computational Areas

Area 1: $\hat{\theta}_1 = \{\theta_{30}, \theta_{66}\} \;\rightarrow\; (\hat{H}_1 = \begin{bmatrix} H_{30} \\ H_{66} \end{bmatrix}, \hat{\mathbf{c}}_1 = \begin{bmatrix} \mathbf{c}_{30} \\ \mathbf{c}_{66} \end{bmatrix})$

Area 2: $\hat{\theta}_2 = \{\theta_{16}, \theta_{53}\} \;\rightarrow\; (\hat{H}_1 = \begin{bmatrix} H_{16} \\ H_{53} \end{bmatrix}, \hat{\mathbf{c}}_1 = \begin{bmatrix} \mathbf{c}_{16} \\ \mathbf{c}_{53} \end{bmatrix})$

Area 3: $\hat{\theta}_3 = \{\theta_{68}\} \;\rightarrow\; (\hat{H}_3 = H_{68}, \hat{\mathbf{c}}_3 = \mathbf{c}_{68})$

Area 4: $\hat{\theta}_4 = \{\theta_{56}\} \;\rightarrow\; (\hat{H}_4 = H_{56}, \hat{\mathbf{c}}_4 = \mathbf{c}_{56})$

Global Optimization Problem:

$$\underset{\mathbf{a}_1, \ldots, \mathbf{a}_N, \mathbf{z}}{\text{minimize}} \sum_{i=1}^{N} \frac{1}{2} \left\| \hat{H}_i \mathbf{a}_i - \hat{\mathbf{c}}_i \right\|_2^2$$

$$\text{subject to } \mathbf{a}_i - \mathbf{z} = 0, \text{ for } i = 1, \ldots, N$$

Solve using

Alternating Direction Method of Multipliers (ADMM)

Distributed Optimization Using ADMM

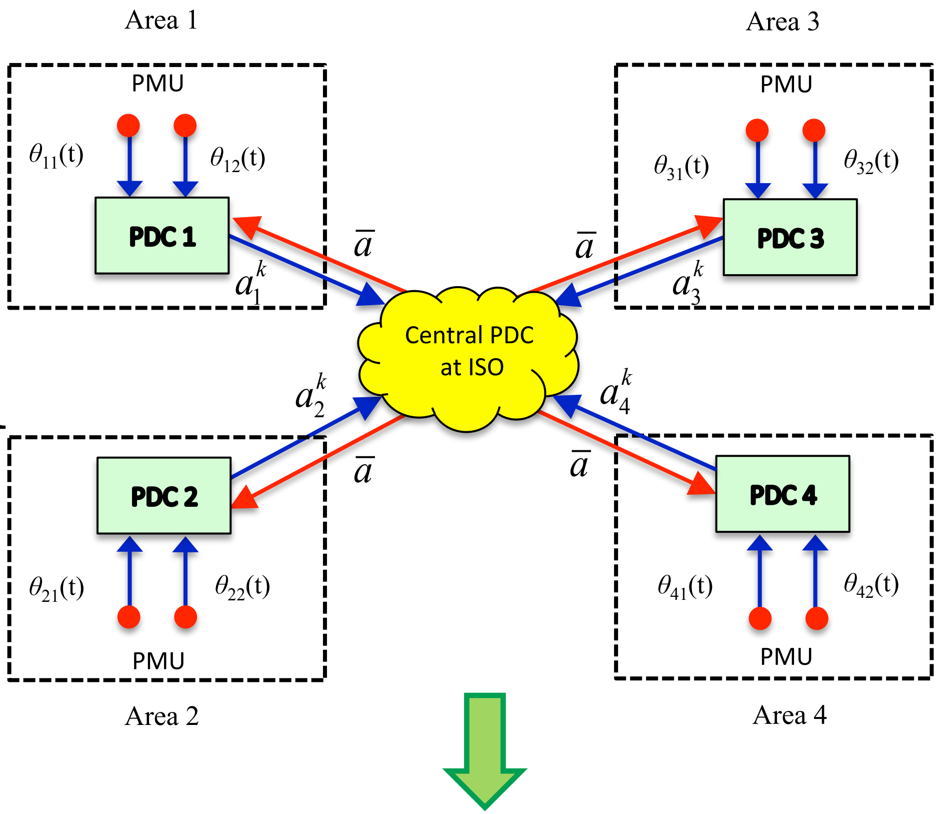
Iteration *k+1*

- Step 1 Update a_i and w_i locally at PDC i

$$\mathbf{a}_i^{k+1} = ((H_i^k)^T H_i^k + \rho I)^{-1}((H_i^k)^T \mathbf{c}_i^k - \mathbf{w}_i^k + \rho \bar{\mathbf{a}}^k)$$

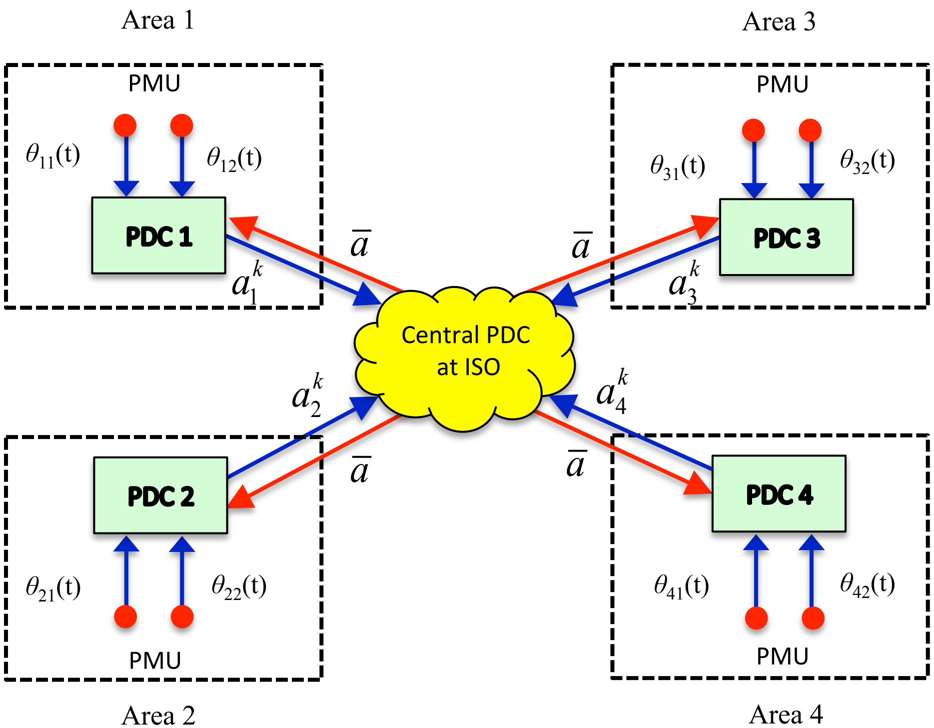
$$\mathbf{w}_i^{k+1} = \mathbf{w}_i^k + \rho(\mathbf{a}_i^{k+1} - \bar{\mathbf{a}}^{k+1})$$

- Step 2 Gather the values of a_i^{k+1} at the central PDC
- Step 3 Take the average of a_i^{k+1}
- Step 4 Broadcast the average value ($\bar{a}_i^{k+1}$) to local PDCs
- Step 5 Check the convergence
- Final Step Find the frequency Ω_i, and damping σ_i at each local PDC using a_i^{k+1}



Privacy of PMU data between companies guaranteed

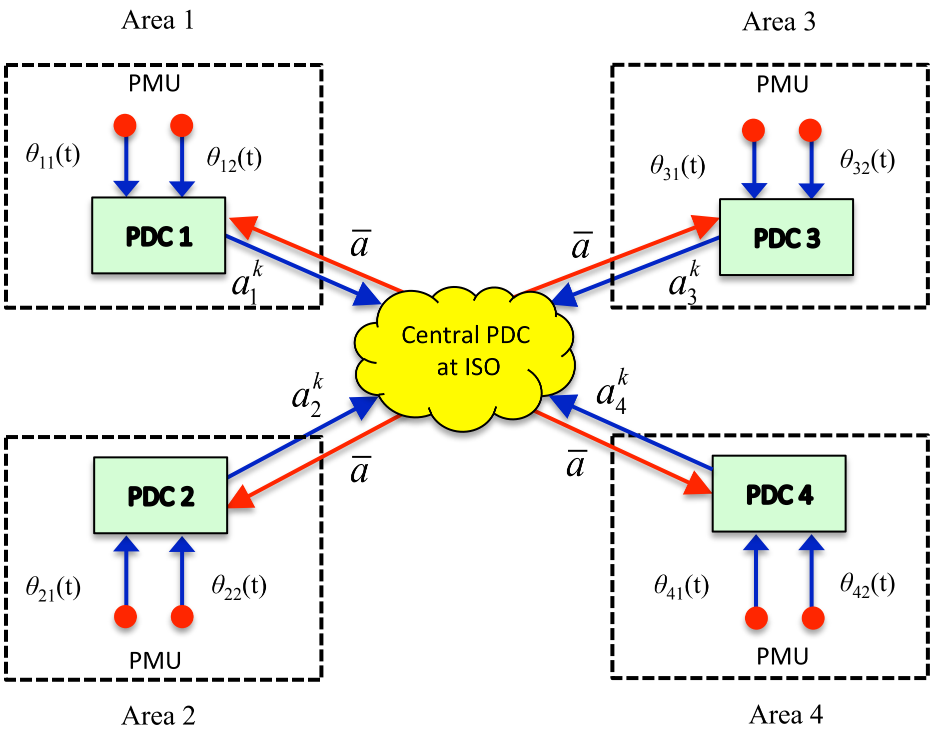
Cyber-Physical Coupling:
Incorporating Asynchronous Wide-Area Communication



Traffic Models for Internet Delays:

$$P(t) = \frac{1}{2}[\text{erf}(\frac{\mu}{\sqrt{2}\sigma}) + \text{erf}(\frac{t-\mu}{\sqrt{2}\sigma})] + \frac{(1-p)}{N} e^{(\frac{1}{2}\lambda^2\sigma^2 + \mu\lambda)} [\text{erf}(\frac{\lambda\sigma^2 + \mu}{\sqrt{2}\sigma}) + \text{erf}(\frac{t - \lambda\sigma^2 - \mu}{\sqrt{2}\sigma})]$$

Cyber-Physical Coupling:
Incorporating Asynchronous Wide-Area Communication



If a message doesn't arrive at ISO by a delay threshold d_1*

- **Strategy 1:**

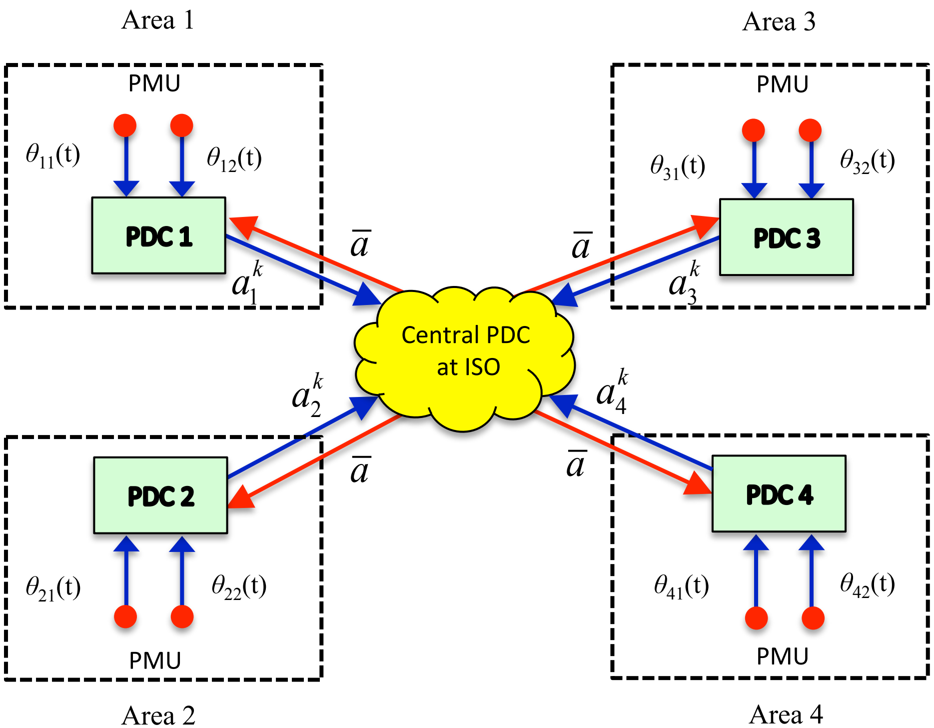
$$z^{(k+1)} = \frac{1}{|S_1^{(k)}|} \sum_{i \in S_1^{(k)}}^{N} \left(a_i^{(k+1)} + \frac{1}{\rho} w_i^{(k)}\right)$$

→ **Can easily lead to divergence**

Traffic Models for Internet Delays:

$$P(t) = \frac{1}{2}\left[\mathrm{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right) + \mathrm{erf}\left(\frac{t-\mu}{\sqrt{2}\sigma}\right)\right] + \frac{(1-p)}{N} e^{\left(\frac{1}{2}\lambda^2\sigma^2 + \mu\lambda\right)}\left[\mathrm{erf}\left(\frac{\lambda\sigma^2 + \mu}{\sqrt{2}\sigma}\right) + \mathrm{erf}\left(\frac{t - \lambda\sigma^2 - \mu}{\sqrt{2}\sigma}\right)\right]$$

Cyber-Physical Coupling:
Incorporating Asynchronous Wide-Area Communication



If a message doesn't arrive at ISO by a delay threshold d_1*

- **Strategy 2:**

$$z^{(k+1)} = \frac{1}{N}\left(\sum_{i \in S_1^{(k)}}^{N} \left(a_i^{(k+1)} + \frac{1}{\rho} w_i^{(k)}\right) + \sum_{i \notin S_1^{(k)}}^{N} \left(a_i^{(k)} + \frac{1}{\rho} w_i^{(k-1)}\right)\right)$$

Substitute values from previous iteration

Convergent, but slow

Traffic Models for Internet Delays:

$$P(t) = \frac{1}{2}\left[\operatorname{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right) + \operatorname{erf}\left(\frac{t-\mu}{\sqrt{2}\sigma}\right)\right] + \frac{(1-p)}{N} e^{\left(\frac{1}{2}\lambda^2\sigma^2 + \mu\lambda\right)} \left[\operatorname{erf}\left(\frac{\lambda\sigma^2 + \mu}{\sqrt{2}\sigma}\right) + \operatorname{erf}\left(\frac{t - \lambda\sigma^2 - \mu}{\sqrt{2}\sigma}\right)\right]$$

Hybrid Update Strategies for A-ADMM

Uplink: Central PDC uses strategies for delayed message from local PDCs

Strategy I: Skipping

$$z^{(k+1)} = \frac{1}{|S_1^{(k)}|} \sum_{i \in S_1^{(k)}} (a_i^{(k+1)} + (1/\rho) w_i^{(k)})$$

Strategy II: Using Previous Messages

$$z^{(k+1)} = \frac{1}{N} \Big(\sum_{i \in S_1^{(k)}} (a_i^{(k+1)} + (1/\rho) w_i^{(k)}) + \sum_{i \notin S_1^{(k)}} (a_i^{(l_i+1)} + (1/\rho) w_i^{(l_i)}) \Big)$$

$l_i \in (k-1, k-2, \ldots)$ index of the latest message that arrived at the central PDC for i^{th} local PDC

Strategy II with Gradient Method

$$z^{(k+1)} = \frac{1}{N} \Big(\sum_{i \in S_1^{(k)}} (a_i^{(k+1)} + (1/\rho) w_i^{(k)}) + \sum_{i \notin S_1^{(k)}} (a_i^{(u_i)} + \beta_i (a_i^{(u_i)} - a_i^{(u_i-1)}) + (1/\rho) w_i^{(u_i)}) \Big)$$

Downlink: Each local PDC uses strategies for delayed message from central PDC

Strategy I: Skipping

Retransmits the previous local updates to the central PDC

Strategy II: Using Previous Messages

$$w_i^{(k)} = w_i^{(k-1)} + \rho (a_i^{(k)} - z^{(l_i)})$$

$$a_i^{(k+1)} = ((\hat{H}_i^{(k)})^T \hat{H}_i^{(k)} + \rho I)^{-1} ((\hat{H}_i^{(k)})^T \hat{c}_i^{(k)} - w_i^{(k)} + \rho z^{(l_i)})$$

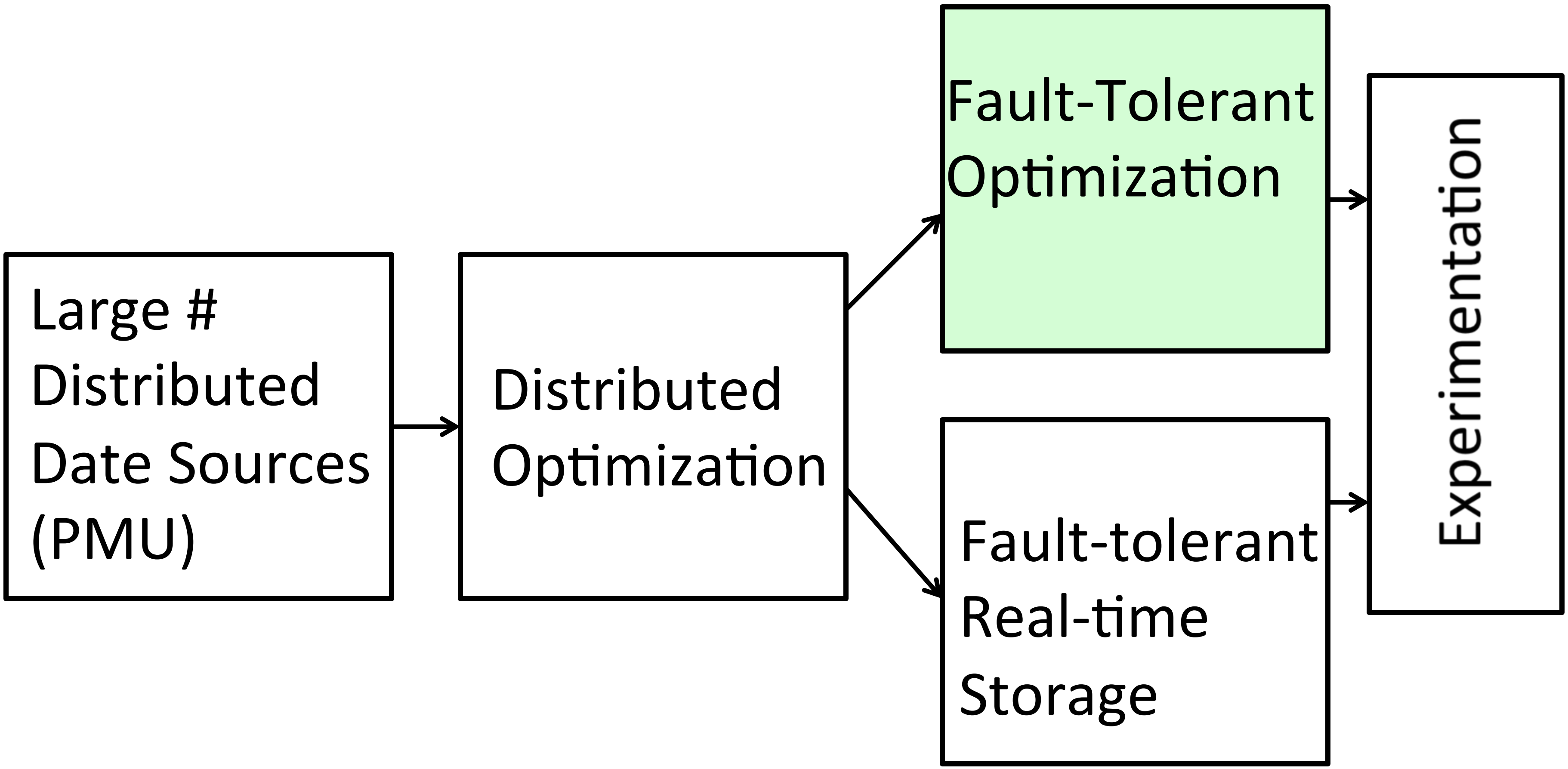
$l_i \in (k-1, k-2, \ldots)$ index of the latest message that arrived at the i^{th} local l PDC

Strategy II with Gradient Method

$$w_i^{(k)} = w_i^{(k-1)} + \rho (a_i^{(k)} - (z^{(l_i)} + \gamma_i (z^{(l_i)} - z^{(l_i-1)})))$$

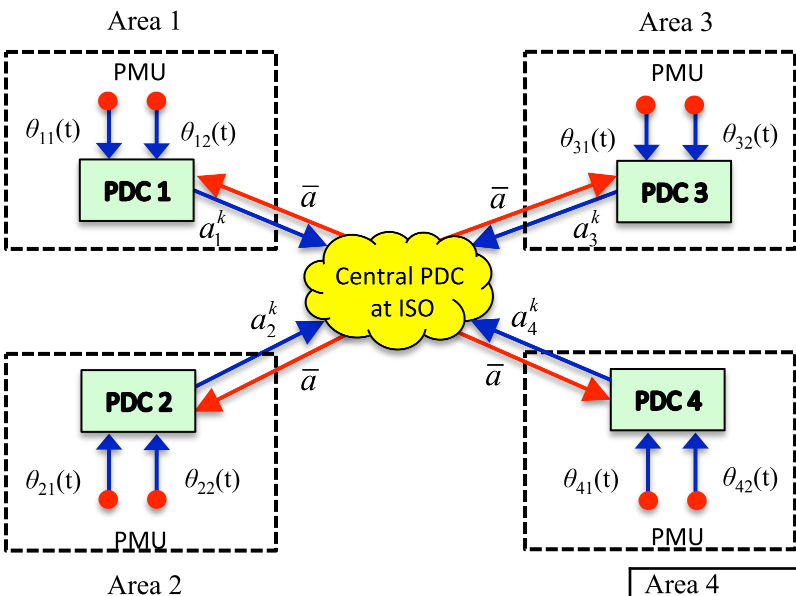
$$a_i^{(k+1)} = ((\hat{H}_i^{(k)})^T \hat{H}_i^{(k)} + \rho I)^{-1} ((\hat{H}_i^{(k)})^T \hat{c}_i^{(k)} - w_i^{(k)} + \rho (z^{(l_i)} + \gamma_i (z^{(l_i)} - z^{(l_i-1)})))$$

Outline


Large #
Distributed
Date Sources
(PMU)
Distributed
Optimization
Fault-Tolerant
Optimization
Fault-tolerant
Real-time
Storage
Experimentation

Wide-Area Oscillation Estimation

Distributed:



Multiple Computational Areas

$$\text{Area 1}: \ \hat{\theta}_1 = \{\theta_{30}, \theta_{66}\} \ \rightarrow \ (\hat{H}_1 = \begin{bmatrix} H_{30} \\ H_{66} \end{bmatrix}, \hat{\mathbf{c}}_1 = \begin{bmatrix} \mathbf{c}_{30} \\ \mathbf{c}_{66} \end{bmatrix})$$

$$\text{Area 2}: \ \hat{\theta}_2 = \{\theta_{16}, \theta_{53}\} \ \rightarrow \ (\hat{H}_1 = \begin{bmatrix} H_{16} \\ H_{53} \end{bmatrix}, \hat{\mathbf{c}}_1 = \begin{bmatrix} \mathbf{c}_{16} \\ \mathbf{c}_{53} \end{bmatrix})$$

$$\text{Area 3}: \ \hat{\theta}_3 = \{\theta_{68}\} \ \rightarrow \ (\hat{H}_3 = H_{68}, \hat{\mathbf{c}}_3 = \mathbf{c}_{68})$$

$$\text{Area 4}: \ \hat{\theta}_4 = \{\theta_{56}\} \ \rightarrow \ (\hat{H}_4 = H_{56}, \hat{\mathbf{c}}_4 = \mathbf{c}_{56})$$

Global Optimization Problem:

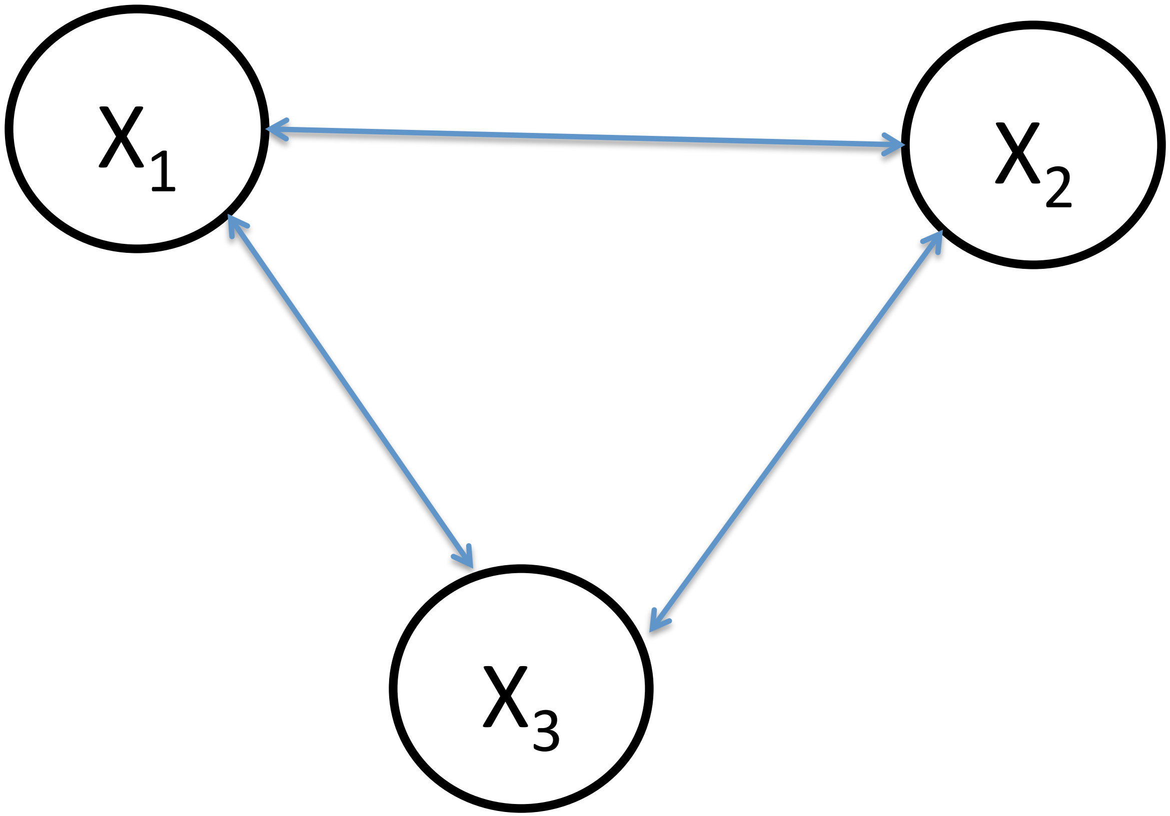
$$\underset{\mathbf{a}_1, K, \mathbf{a}_N, \mathbf{z}}{\text{minimize}} \sum_{i=1}^{N} \frac{1}{2} \left\| \hat{H}_i \mathbf{a}_i - \hat{\mathbf{c}}_i \right\|_2^2$$

$$\text{subject to } \mathbf{a}_i - \mathbf{z} = 0, \text{ for } i = 1, \ldots, N$$

Distributed Optimization

Node i has local objective $h_i(x)$

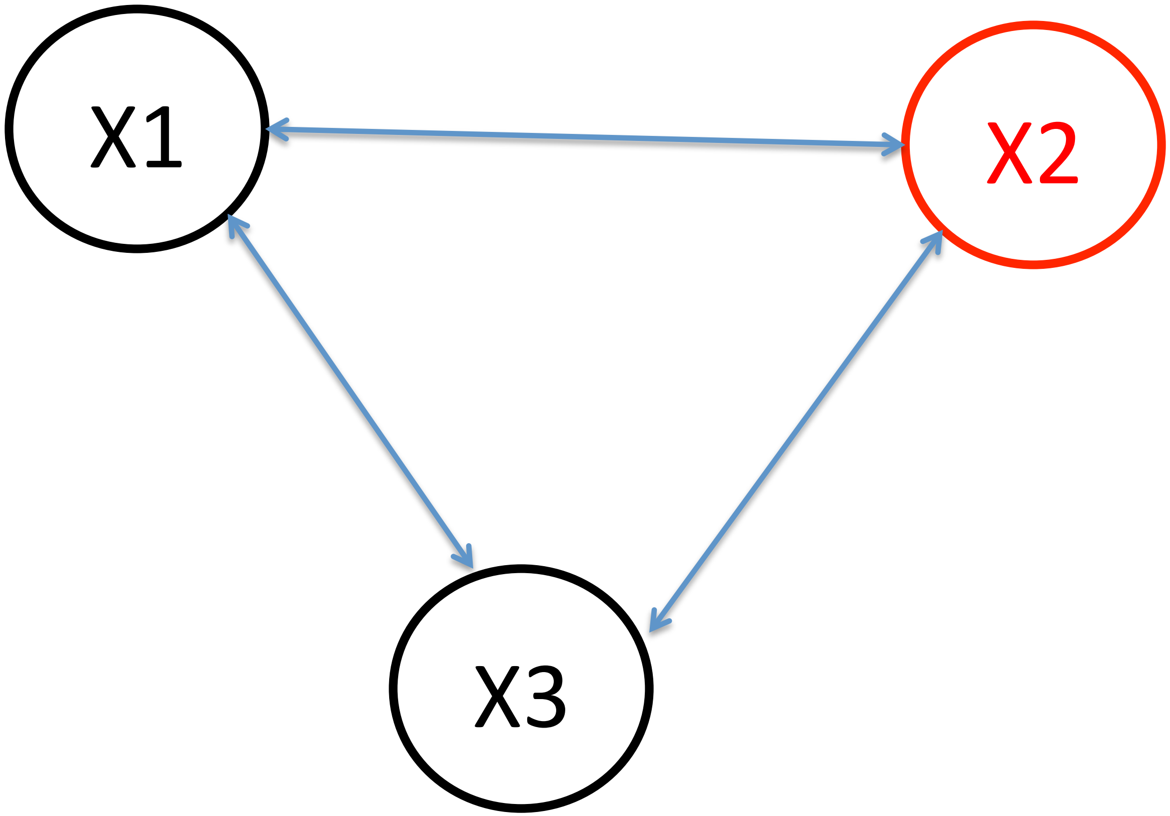
minimize $$h(x) = \frac{1}{n}\sum_{i=1}^{n} h_i(x)$$



$$X_3 \leftarrow \tfrac{1}{3}(X_1+X_2+X_3) - \lambda_t \, \text{grad}\, h_3(X3)$$

Many Other Applications

- Distributed robotics
- Machine learning



$X3 \leftarrow \frac{1}{3}(X1+X2+X3) - \text{grad}\, h_3(X3)$

Fault-Tolerance

Not meaningful to optimize

$$h(x) = \frac{1}{n}\sum_{i=1}^{n} h_i(x)$$

since faulty costs included

Alternative Goal

N = non-faulty nodes

Optimize non-faulty cost functions:

$$h(x) = \frac{1}{|N|} \sum_{i \in N} h_i(x).$$

... but this is provably impossible

Byzantine Fault-Tolerant Optimization

Instead of uniform weights in

$$h(x) = \frac{1}{|N|} \sum_{i \in N} h_i(x).$$

allow unequal weights

$$h(x) = \sum_{i \in N} \alpha_i \, h_i(x),$$

… but as close to uniform as possible

Byzantine Fault-Tolerant Optimization

- Optimal algorithms
  - How many weights non-zero?
  - How large can they be?

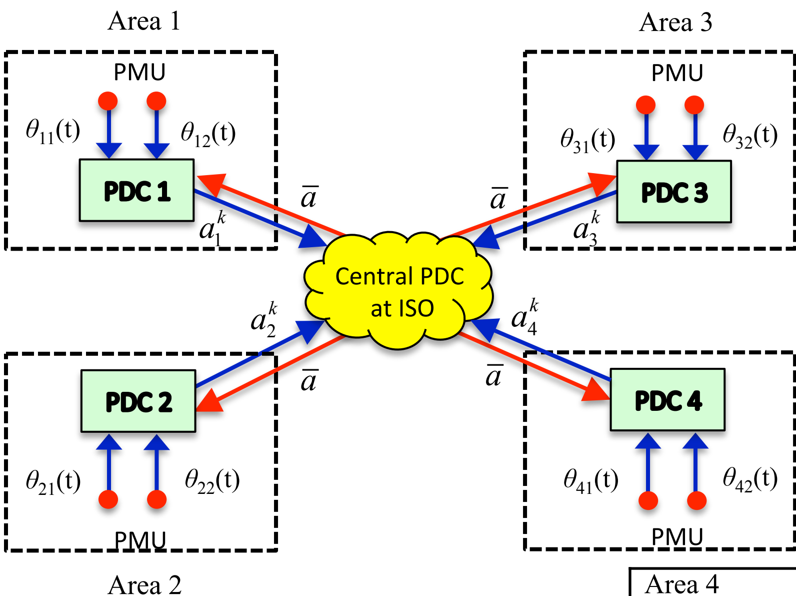
$$h(x) = \sum_{i \in N} \alpha_i \, h_i(x),$$

Byzantine Fault-Tolerant Optimization

- Optimal algorithms for complete networks
- Many related problems open …

Wide-Area Oscillation Estimation

Distributed:



Multiple Computational Areas

$$\text{Area 1: } \hat{\theta}_1 = \{\theta_{30}, \theta_{66}\} \rightarrow (\hat{H}_1 = \begin{bmatrix} H_{30} \\ H_{66} \end{bmatrix}, \hat{\mathbf{c}}_1 = \begin{bmatrix} \mathbf{c}_{30} \\ \mathbf{c}_{66} \end{bmatrix})$$

$$\text{Area 2: } \hat{\theta}_2 = \{\theta_{16}, \theta_{53}\} \rightarrow (\hat{H}_1 = \begin{bmatrix} H_{16} \\ H_{53} \end{bmatrix}, \hat{\mathbf{c}}_1 = \begin{bmatrix} \mathbf{c}_{16} \\ \mathbf{c}_{53} \end{bmatrix})$$

$$\text{Area 3: } \hat{\theta}_3 = \{\theta_{68}\} \rightarrow (\hat{H}_3 = H_{68}, \hat{\mathbf{c}}_3 = \mathbf{c}_{68})$$

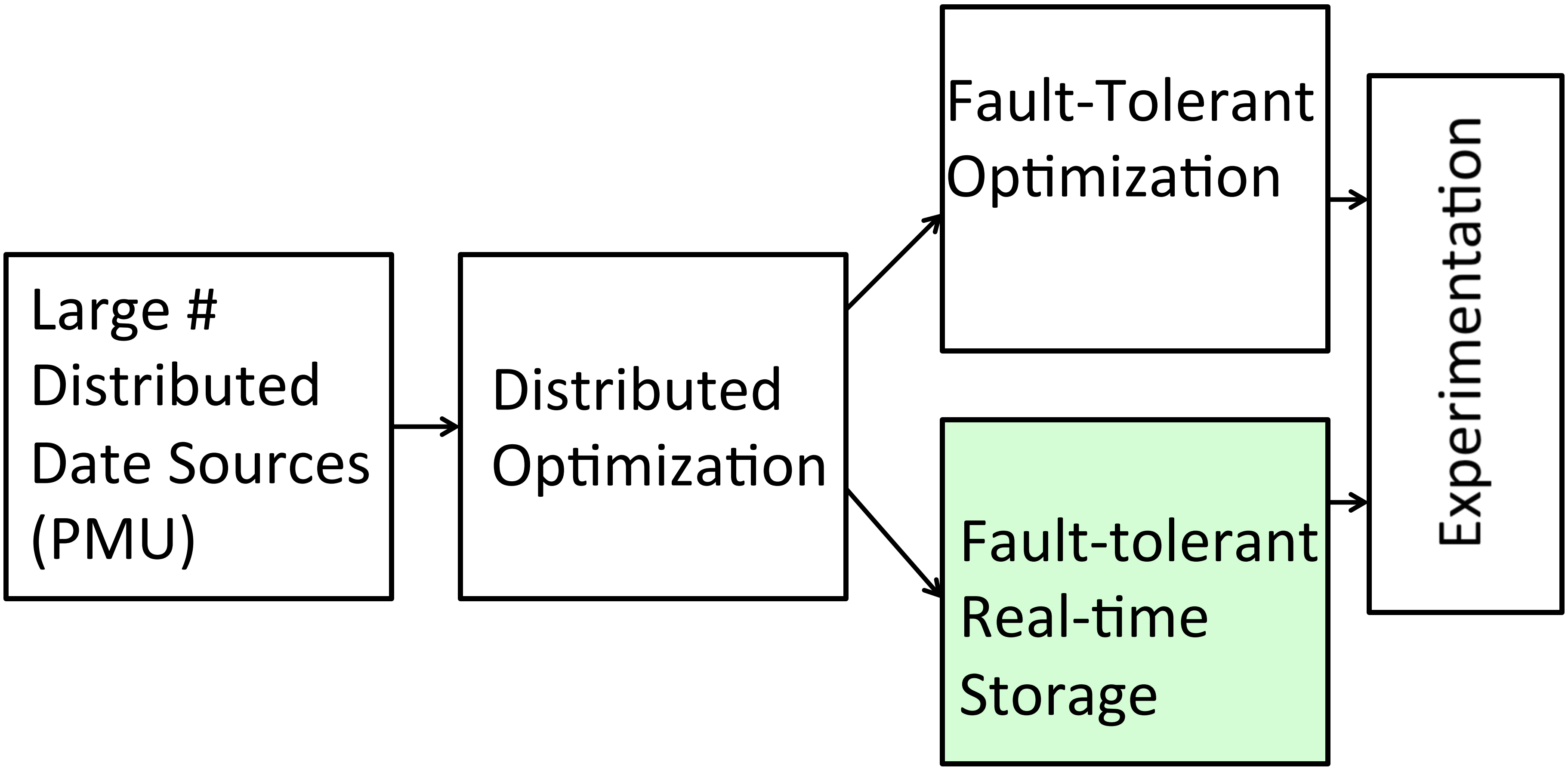
$$\text{Area 4: } \hat{\theta}_4 = \{\theta_{56}\} \rightarrow (\hat{H}_4 = H_{56}, \hat{\mathbf{c}}_4 = \mathbf{c}_{56})$$

Global Optimization Problem:

$$\underset{\mathbf{a}_1,\ldots,\mathbf{a}_N,\mathbf{z}}{\text{minimize}} \sum_{i=1}^{N} \frac{1}{2} \left\| \hat{H}_i \mathbf{a}_i - \hat{\mathbf{c}}_i \right\|_2^2$$

$$\text{subject to } \mathbf{a}_i - \mathbf{z} = 0, \text{ for } i = 1, \ldots, N$$

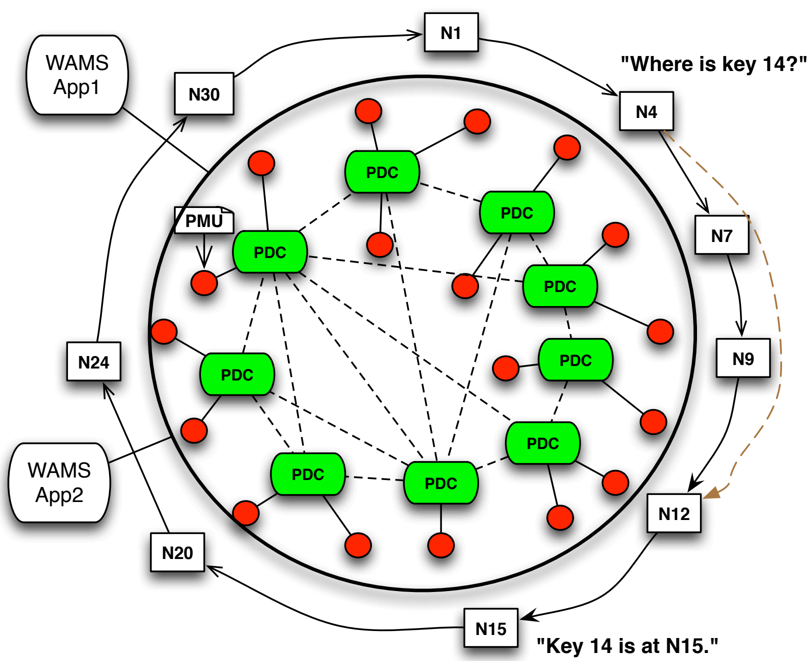
Outline

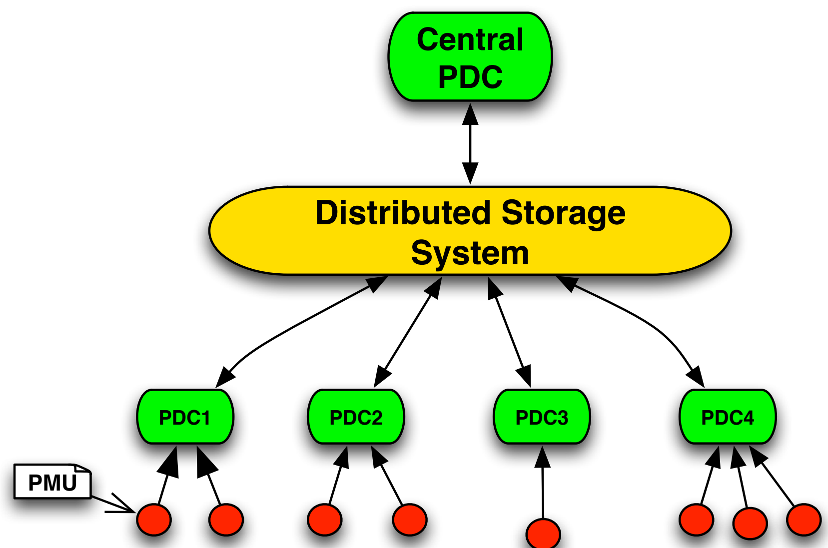

Large #
Distributed
Date Sources
(PMU)
Distributed
Optimization
Fault-Tolerant
Optimization
Fault-tolerant
Real-time
Storage
Experimentation

Resilient Real Time Data Middleware

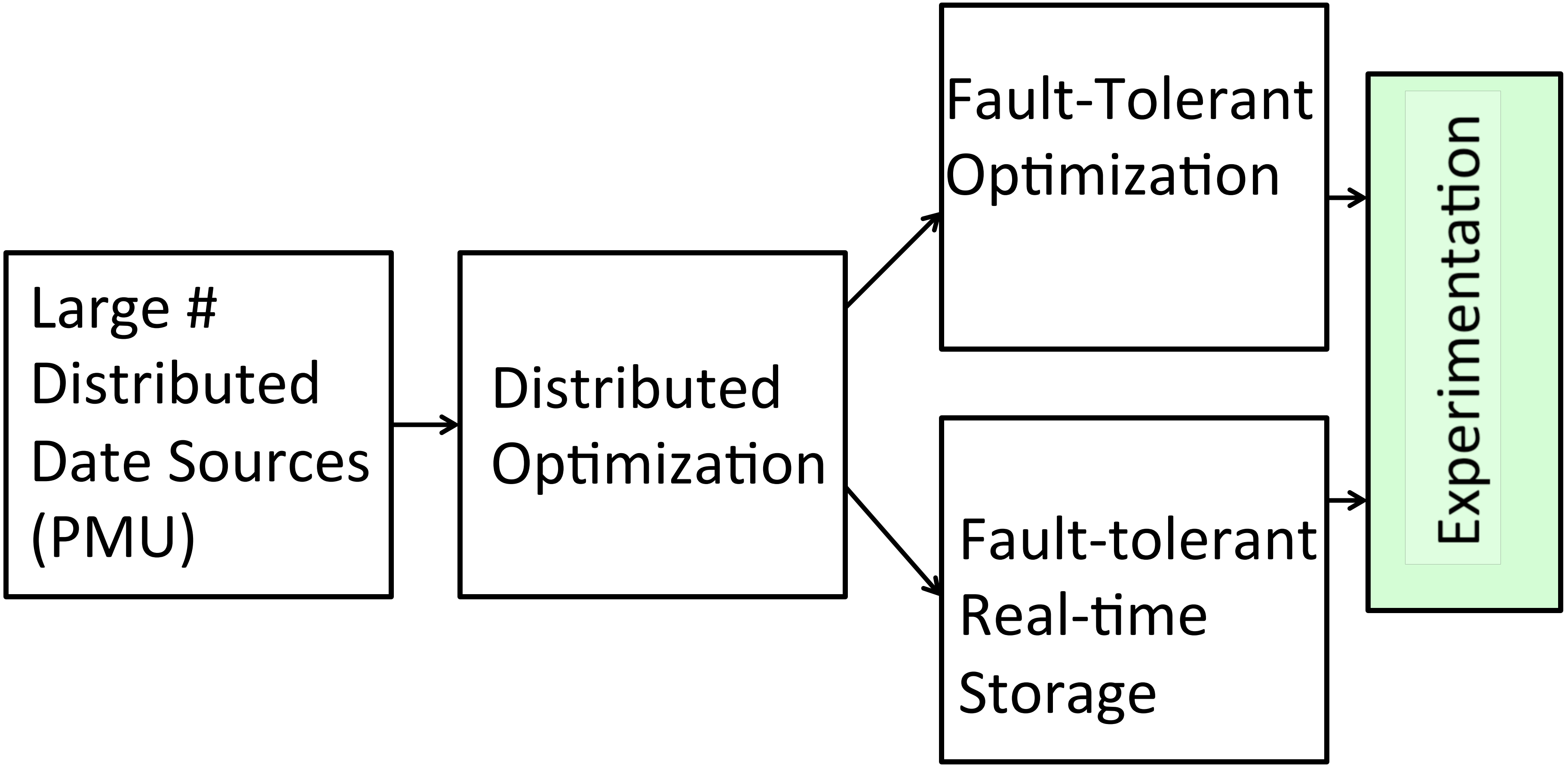
RT-DHT: real-time distributed hash table

- Decouple strong dependency between PDCs and PMU sources
- Chord-like ring + finger pointers
- multiple replicas of data → faults OK
- Network control → deterministic wide –area networks

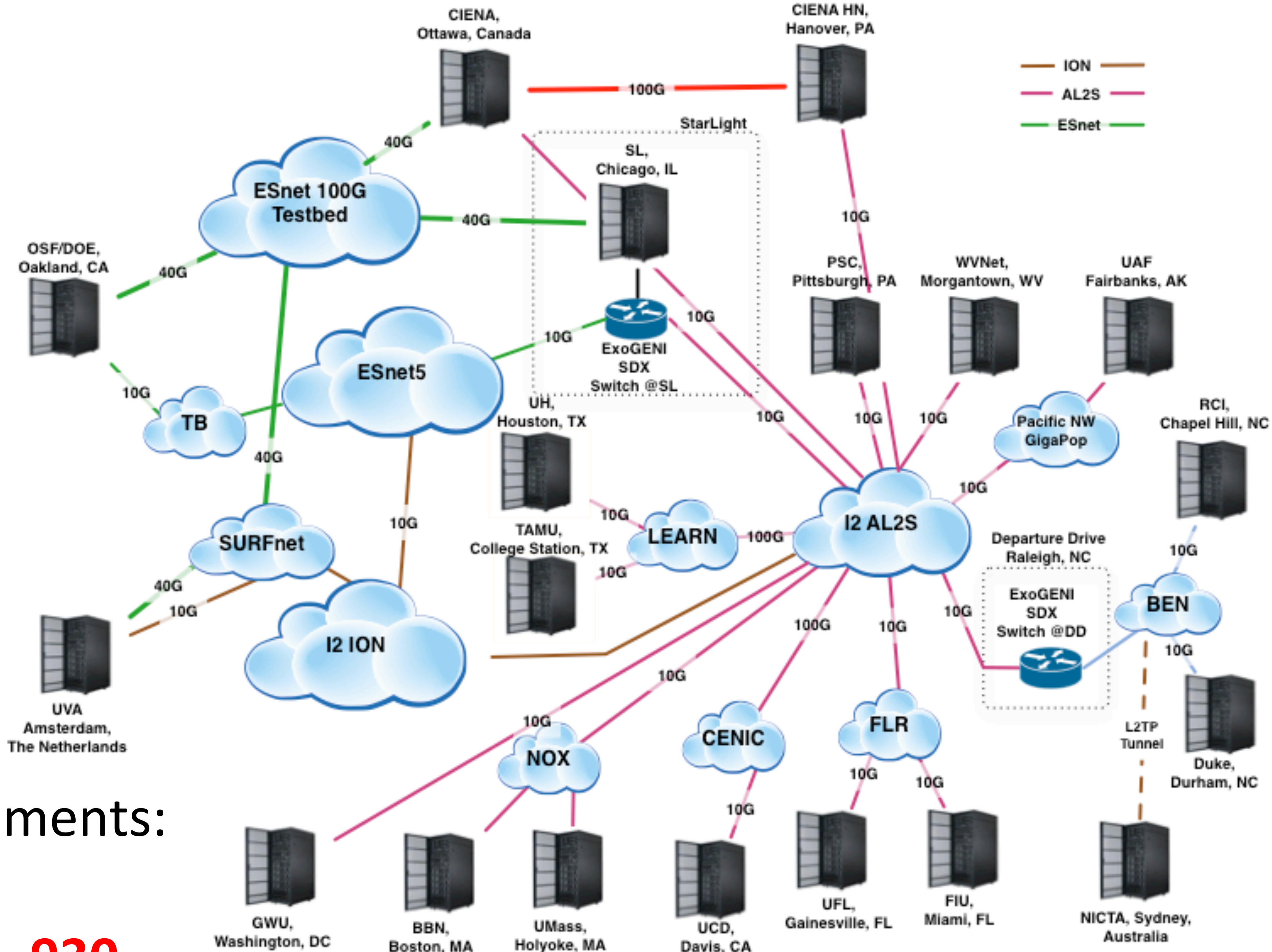




Outline


Large #
Distributed
Date Sources
(PMU)
Distributed
Optimization
Fault-Tolerant
Optimization
Fault-tolerant
Real-time
Storage
Experimentation

ExoGeni TestBed



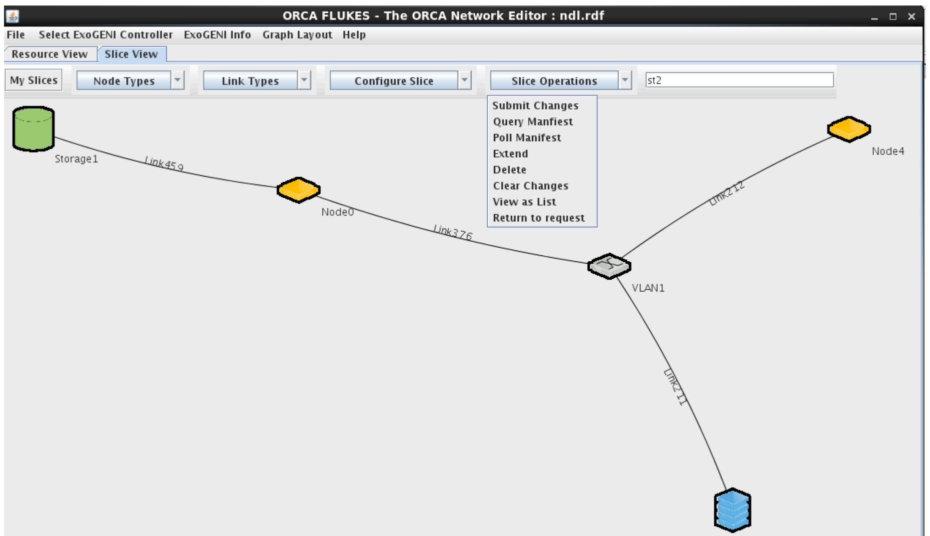
#Experiments:

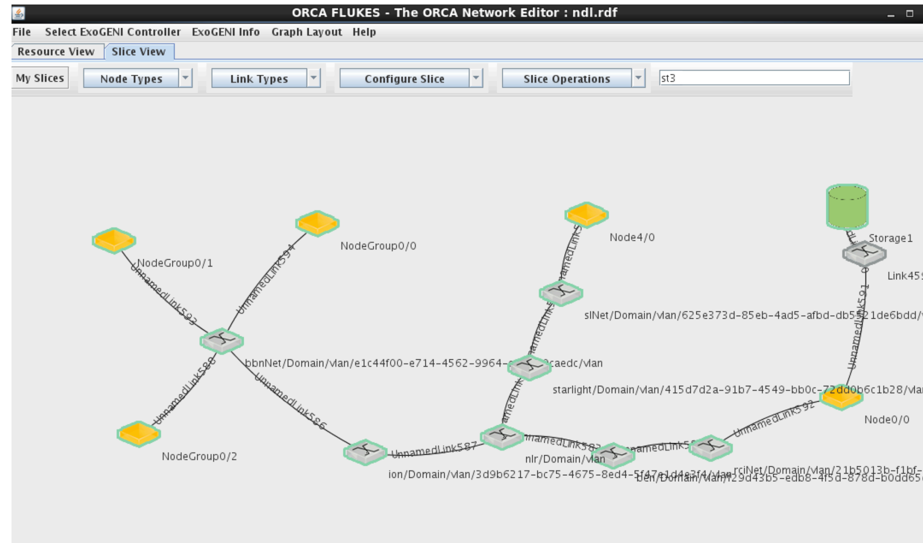
31564

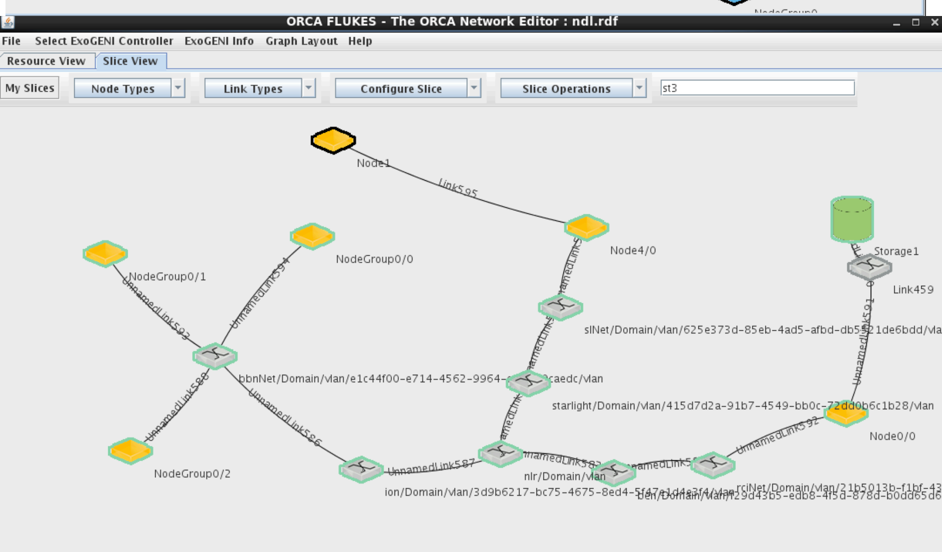
#Users: **930**

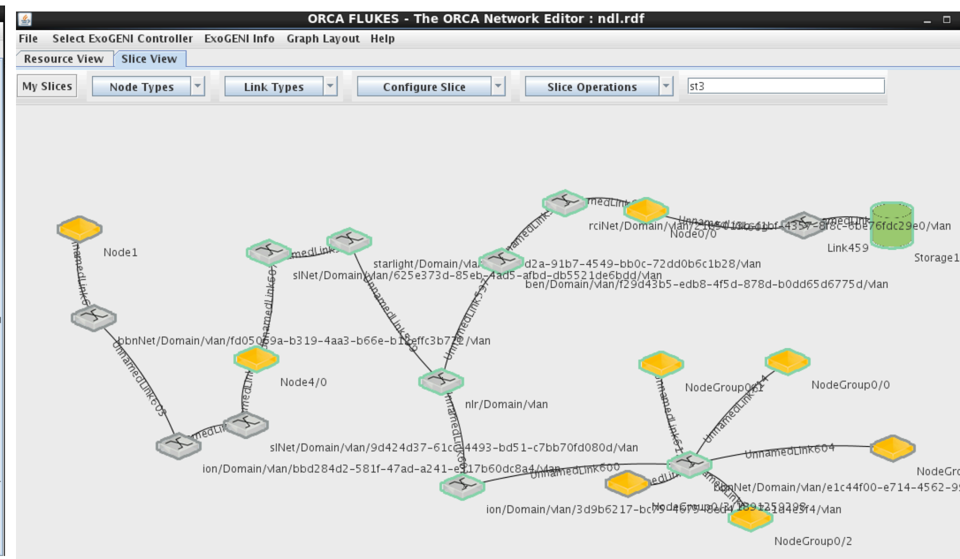
Virtual Networked System: Provisioning, Recovering, and Modifying

- ➢ Create customized OS image for Virtual Machines and C source code for algorithms.
- ➢ Create virtual network topologies on ExoGENI using a web-start app Flukes or GENI tools
- ➢ VM, Baremetal, storage, P2P or Multicasting networks

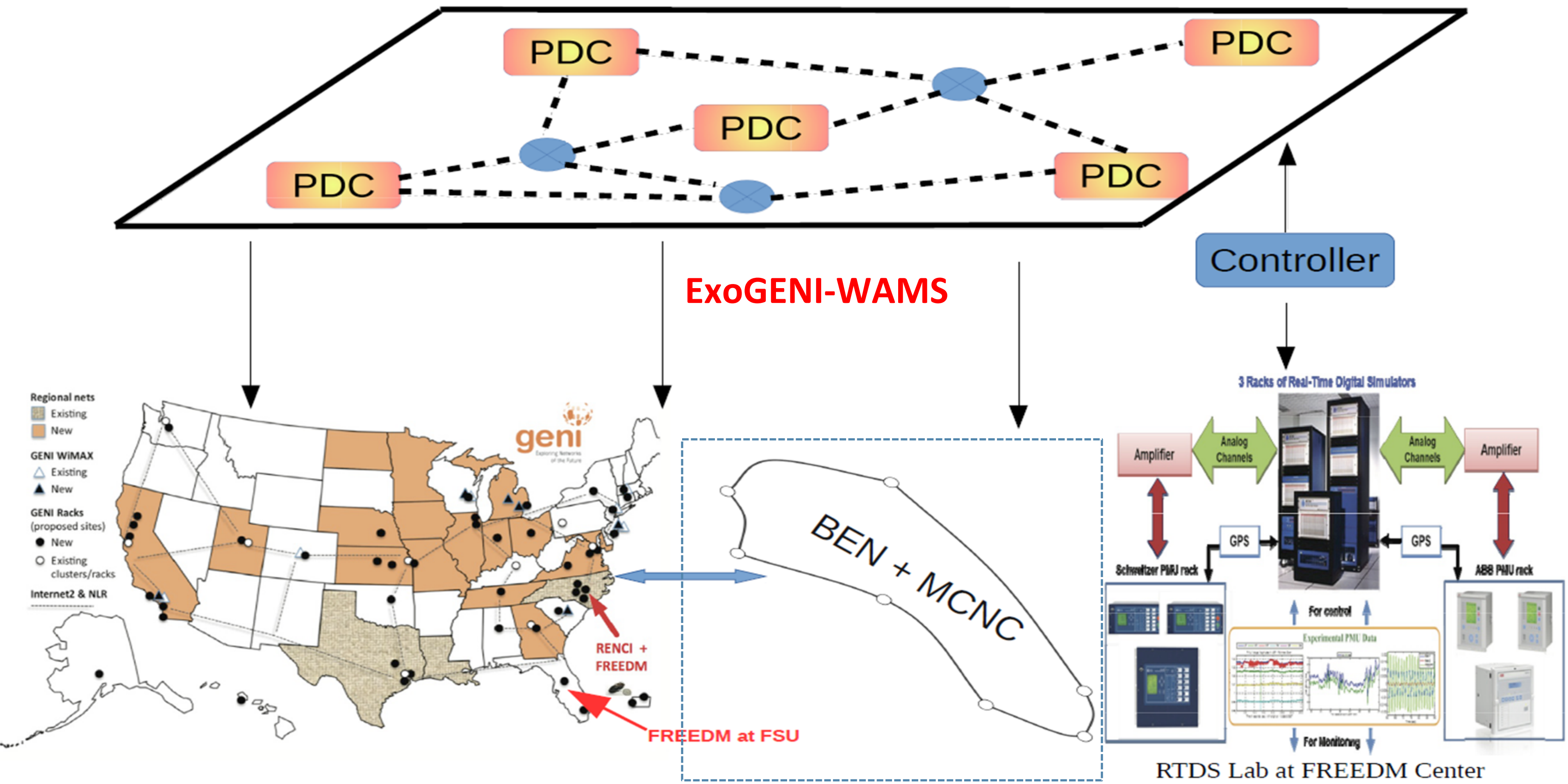








ExoGENI-WAMS Testbed at NC State & RENCI/UNC Chapel Hill



Middleware is being developed currently by Green Energy Corporation and RTI

Thank You