Collaborative Autonomy Work in E-Program

Skynet & Improvements to Distributed Iterative Linear Solvers

Alyson Fox

March 2, 2020

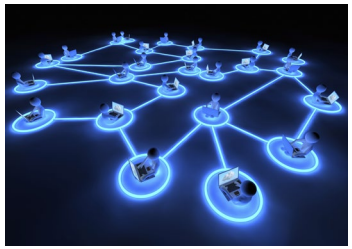
LLNL-PRES-806067

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

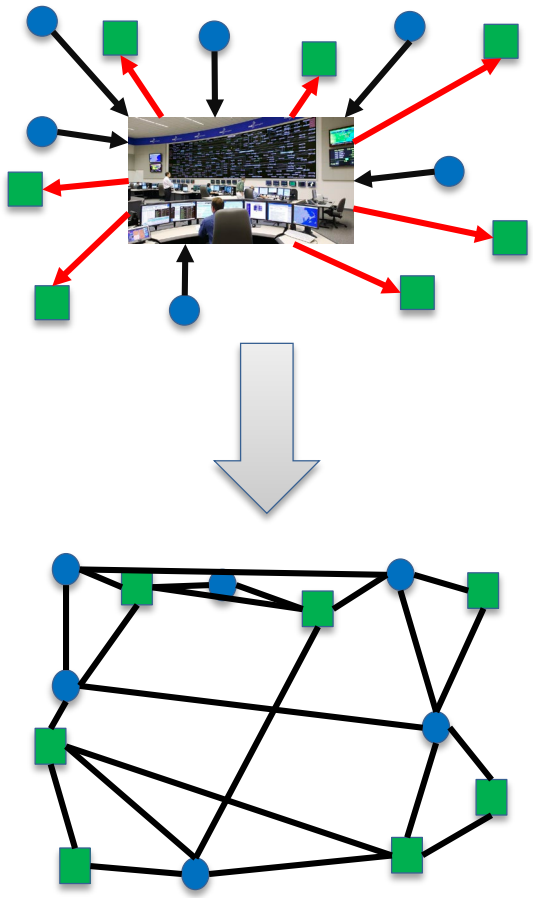
Computing in an Unreliable Environment

- In many real-world settings, we need to perform computation on hardware that is **not reliable.** This hardware could
  - Be delayed, fail, suffer a cyber intrusion, etc.
- Especially true in sensor-based control systems.
  - Power grids
  - Gas pipelines
  - Drone swarms
- In that sort of an unreliable environment, **without even a reliable central controller**, how do you enable *computations* that are reliable?



Collaborative Autonomy

A class of computational methods that enables groups of decentralized devices to work together autonomously to adapt around instabilities in order to *create reliable systemic computation out of unreliable hardware.*



- **Decentralized devices:** A cluster of (usually) small computers with no control center.
- **Autonomous:** Without direction from a human or central controller.
- **Reliable Systemic Computation:** The system as a whole can withstand problems in any given device (or group of devices) and still perform its task.

Collaboratively Autonomy is Common in Nature

- Flocks of birds.
- Ant colonies.
- Animal herds.
- Multicellular biology.

- In all of these systems, damaging or removing one (or several) individuals doesn't compromise the whole, which is able to act as a unit to make decisions and take actions.

The Skynet Software Architecture

A software platform to enable collaborative autonomy

- Skynet is a library of techniques that can be applied across a wide range of collaborative autonomy problems
- It provides computational building blocks that have strong resilience properties, enabling programs built with those blocks to have that same resilience.
- Also provides an easy-to-use interface so that the user does not need to be a collaborative autonomy expert.

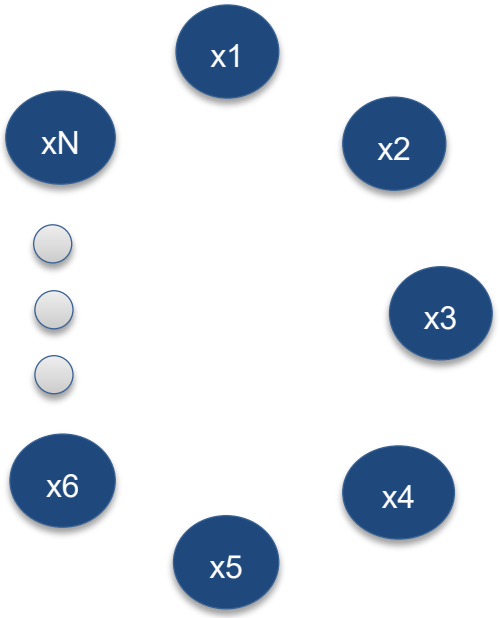
In other words…….it is a
Numerical Linear Algebra Package for Unreliable Computing Environments

Why Do We Need Collaborative Autonomy?

When *no single component* is reliable, even simple operations become nontrivial.

EXAMPLE: Decentralized Averaging

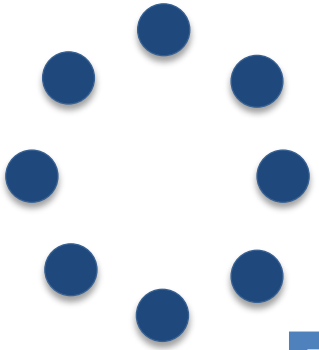
- Every agent (e.g. smart sensor) has a piece of data to include in the average.
- Every agent not malfunctioning should have the answer.
- Which agent does the actual calculation? Is it shared somehow? What kind of communication is needed?

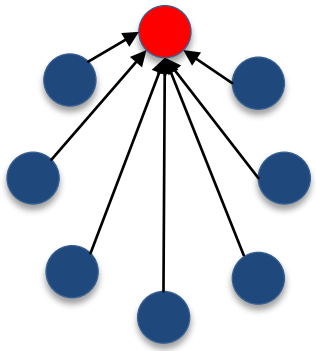


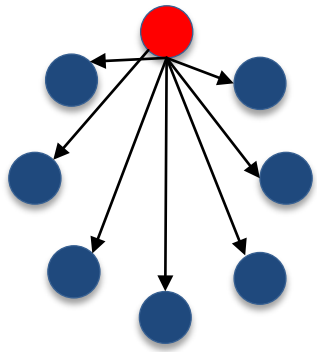
Is the technique robust to…
Device failure
Network link failure
Scalability
Calculation compromise
Source data compromise
Network Knowledge Needs

EXAMPLE: Decentralized Averaging

- Which device does the actual calculation? Is it a group of devices? What kind of communication is needed?
- **Simplest idea:** Send to some device, perform average, broadcast.



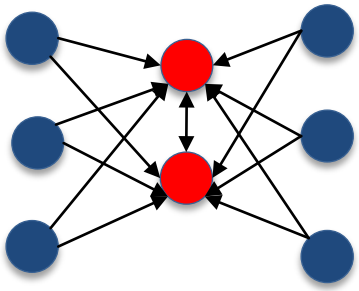




Robust to…	
Device failure	
Network link failure	
Scalability	
Calculation compromise	
Source data compromise	
Network Knowledge Needs	

Decentralized Averaging

- **Another idea:** Send to *two* devices, replicate computation?

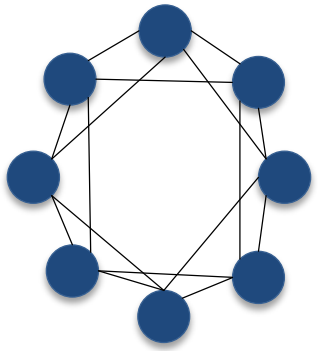


Robust to…	
Device failure	
Network link failure	
Scalability	
Calculation compromise	
Source data compromise	
Network Knowledge Needs	

Decentralized Averaging, Push-Sum Consensus Olshevky et. al. (2018)

- Idea of Push Sum Method
  - **Initialization**: Each agent has its own value and a weight

  Iterate Till Converged:
  - **Communication**: Each agent broadcasts its current value of the average and weight to all local neighbors
  - **Update**: Each agent upd average using informatic

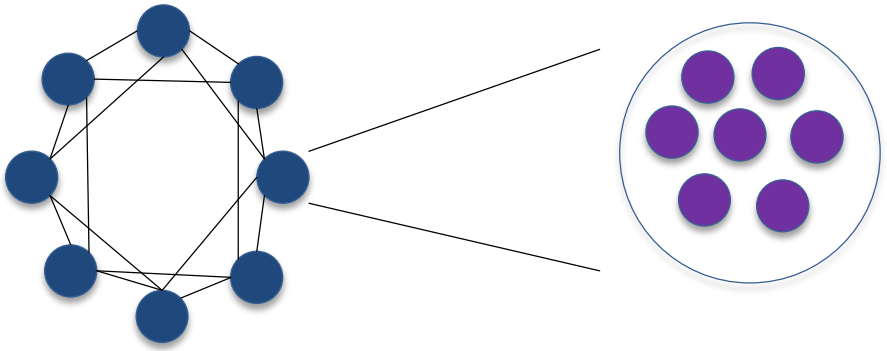


- **Resiliency: Use counte total mass sent by itse mitigate connection/d**

Robust to…	
Device failure	
Network link failure	
Scalability	
Calculation compromise	
Source data compromise	
Network Knowledge Needs	

Decentralized Averaging, Push-Sum Consensus with Calculation Validation

- Every “node” is actually *k* physical agents replicating computation, recipients verify.



Robust to…	
Device failure	
Network link failure	
Scalability	
Calculation compromise	
Source data compromise	
Network Knowledge Needs	

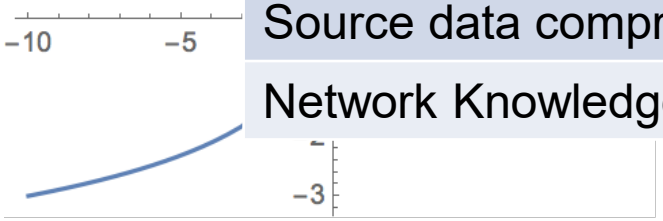
Validating Source Data: How?

- If you know *nothing* about your data, Jon Snow, you cannot validate the data.
- In real applications, you can draw from context:
  - Historical context.
  - Systemic context.
  - Domain context.
- It is almost always possible to design a data validation procedure using context.
  - Can vary in sophistication.
  - Strong systemic and historical context can enable automatic machine learning methods.
- There is no one-size-fits-all procedure.

Decentralized Averaging, Push-Sum Consensus with Calculation Validation and Outlier-Robust Averaging

- **Our contribution:** Apply $f^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} f(x_i)\right)$, where $f(x) = \arcsin(x)$ to mitigate effects of outliers with calculation replication.
- Doesn't alter "normal" data, minimizes impact of outliers. Trying to significa
unreasc
- Locatio

Robust to…	
Device failure	
Network link failure	
Scalability	
Calculation compromise	
Source data compromise	
Network Knowledge Needs	



Current Work on Iterative Linear Solvers

- Many more “complicated” tasks often require a solution to a linear system

Solve $\mathbf{Ax} = \mathbf{b}$, where

$\mathbf{A} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ $\mathbf{b} \in \mathbb{R}^{\mathbf{n}}$ $\mathbf{x}^*$ = exact solution

- Traditional Distributed Method
  - Decompose: $\mathbf{A} = \mathbf{M} - \mathbf{N}$
  - Stationary Iterative solver: $\mathbf{x}^{\mathbf{j}+\mathbf{1}} = \mathbf{M}^{-\mathbf{1}}\left(\mathbf{N}\mathbf{x}^{\mathbf{j}} + \mathbf{b}\right)$
  - At each iterate, each compute node solves a smaller liner system and broadcast the solution.
  - Converges to the solution if the spectral radius of $M^{-1}N$ is less than 1

Current Distributed Implementation

- **Synchronous Jacobi:** $M = D(\text{diagonal of A}),\ N = A - D$

First working on variations to resolve these issues!

$$\left(\qquad \right)_{\mathbf{p}\neq\mathbf{i}}$$

Robust to ...	
Device failure	
Network link failure	
Scalability	
Calculation compromise	
Source data compromise	
Network Knowledge Needs	

Asynchronous Redundant Jacobi

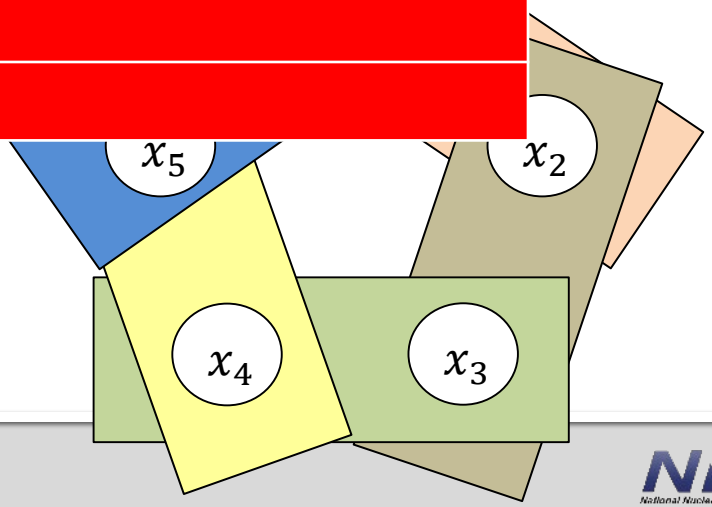
- **Asynchronous Jacobi:**

$$\begin{cases} x_i^j & \text{if } i \neq k_{n+1}(j) \end{cases}$$

Robust to…	
Device failure	
Network link failure	
Scalability	
Calculation compromise	
Source data compromise	
Network Knowledge Needs	

- k … tion
- C…

- Redu…
  - Ea… updating and distributing several components of the solution vector x.



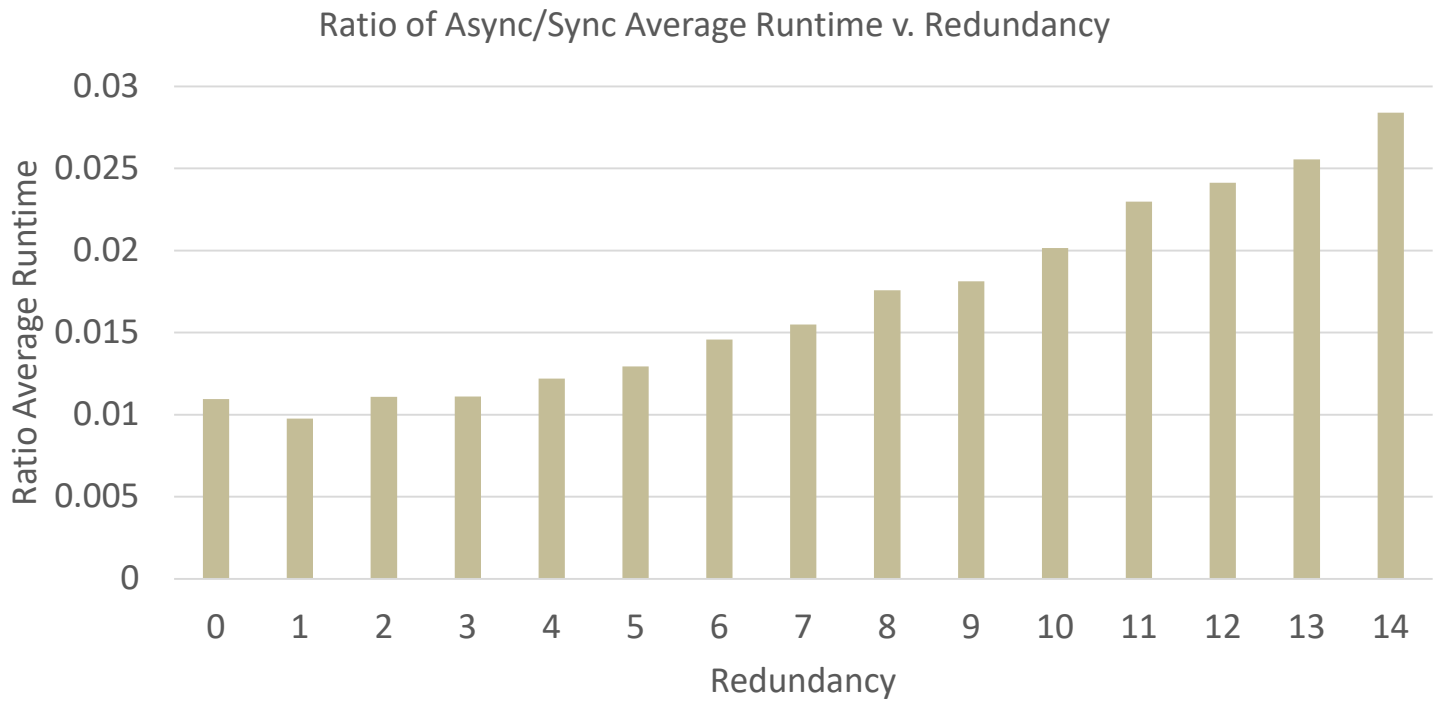
Experiment 1: One process 10x slower than every other process

- **Setup:**

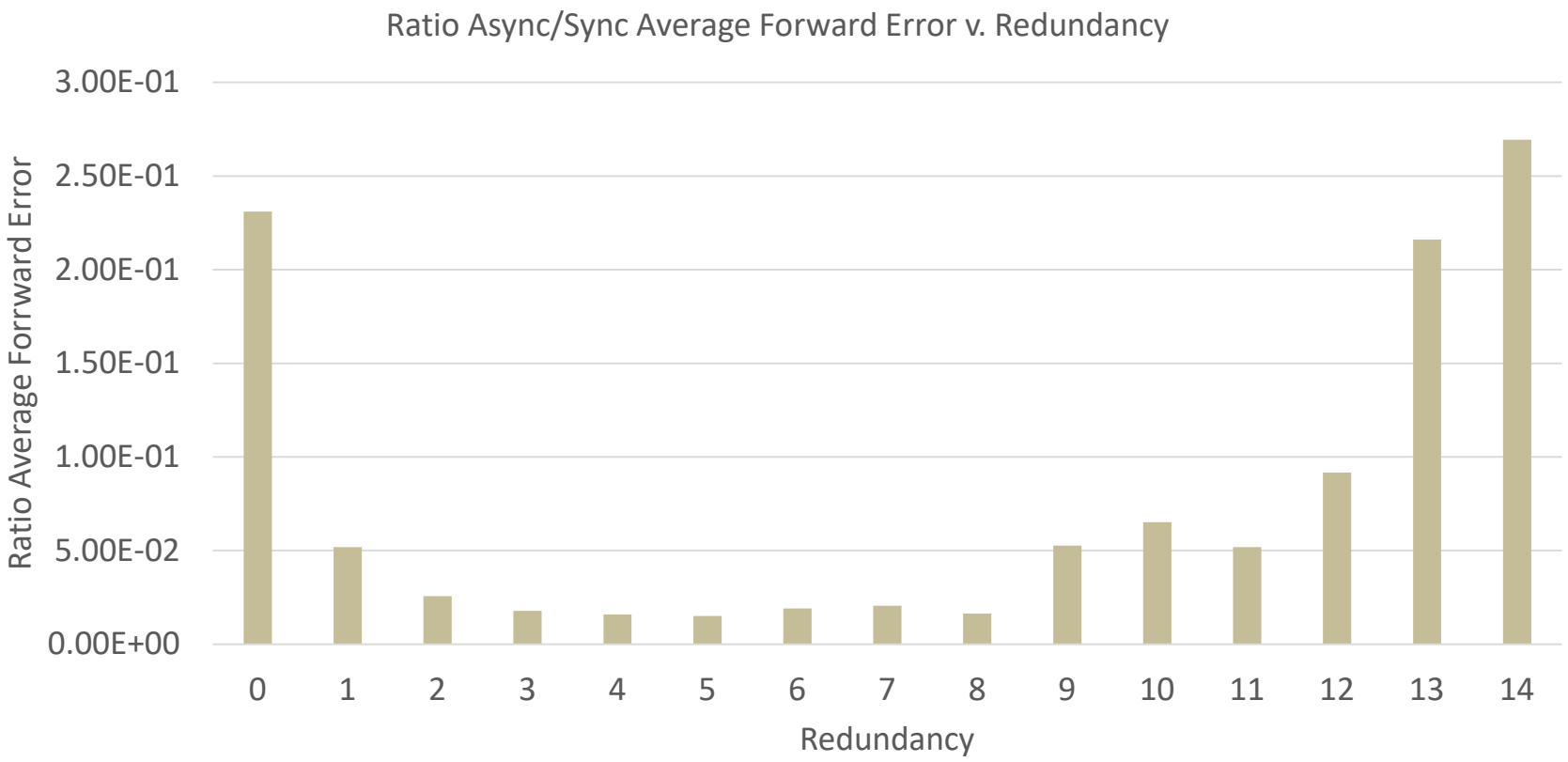
$$\mathbf{A} \in \mathbb{R}^{30\times 30}$$

$$\mathbf{A} = (a_{ij}) = \begin{cases} 1 & \text{if } i \neq j \\ 31 & \text{if } i = j \end{cases}$$

Stopping Criteria: $\dfrac{||x^j - x^*||_2}{||x^*||_2} < 10^{-6}$



Experiment 1: One process is 10x slower than every other process



- Average Forward Error is computed using information from the results of each process: If two or more process output information for the same component of x, the average of that component is used.

Summary

- Skynet will enable users to use CA techniques with minimal effort.
- Current work on robust variants of iterative linear solvers:
  - We can now solve linear systems in unreliable networks where synchronous calculations are infeasible.
  - The results show that the error is significantly better than synchronous Jacobi when facing delays from a single processor => can easily be extrapolated to multiprocessor delays/failures.
- **Future Work:** Further algorithmic work needs to be done to ensure resiliency in classic distributive algorithms.
  - Very few algorithms exist as interest in unreliable computing environment has only recently grown due to the growth of smart devices.

Acknowledgements

- We would like to thank E-program for all the support!
- Skynet /CA Developers:
  - PI: Colin Ponce
  - Adam Harter
  - Chris Vogl
  - Alyson Fox
  - Katie Graham
  - Corey McNeish
- Collaborations on CA Algorithms
  - Aaron Barret & Dr. Agnieszka Miedlar (Kansas University)
- **Funding Projects:** TimeWarp, CS-DERMS, RobustDERMS, Stargazer.

CASC

Center for Applied
Scientific Computing

Lawrence Livermore
National Laboratory