A Scalable Interior Point Solver for Stochastic Programming

Cosmin Petra and Mihai Anitescu

Mathematics and Computer Science Division
Argonne National Laboratory

SP XII Halifax

August 16-20, 2010

MOTIVATION

Optimal management under uncertainty of energy systems

- Applications @Argonne – Anitescu, Constantinescu, Zavala
  - Stochastic Unit Commitment with Wind Power Generation
  - Energy management of Co-generation
  - Economic Optimization of a Building Energy System

- Sources of large dimensional uncertainty in complex energy systems
  - Weather
  - Consumer Demand
  - Market prices

Economic Optimization of a Building Energy System

- **Proactive** management - **temperature forecasting** & **electricity prices**
- Minimize daily energy costs (SAA)

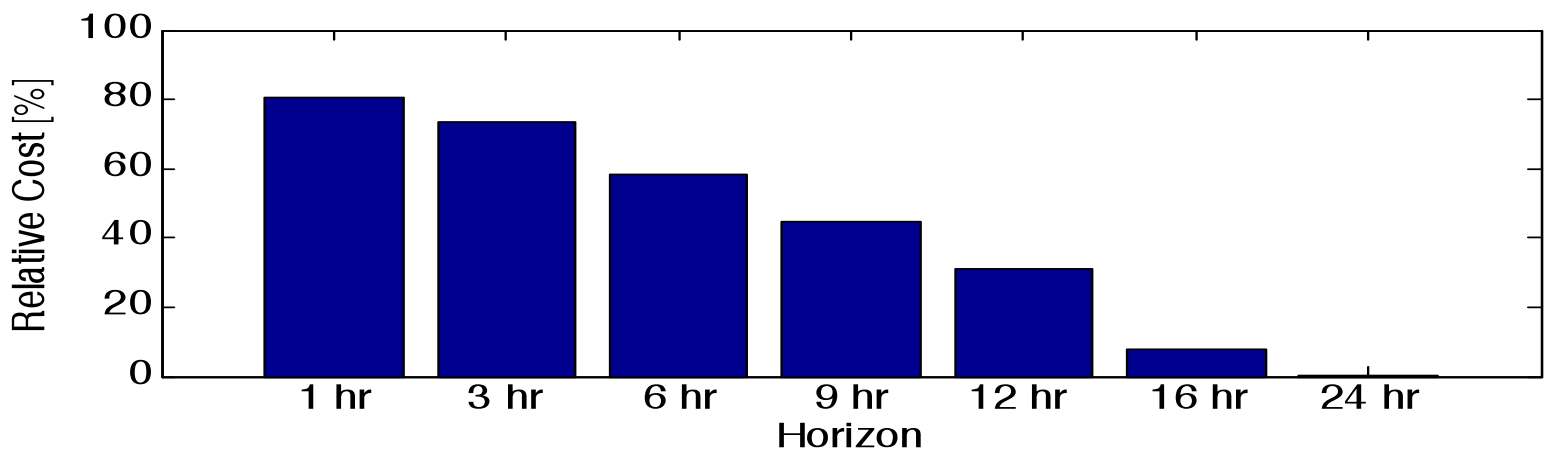
$$\min \quad \text{COST} = \frac{1}{S}\sum_{j=1}^{S}\int_{t}^{t+T}\left[C_{elec}(\tau)\phi_c^{elec}(\tau)+C_{elec}(\tau)\phi_h^{elec}(\tau)+C_{gas}\phi_h^{gas}(\tau)\right]d\tau$$

$$\text{s.t.} \quad C_I\frac{\partial T_I^j}{\partial \tau}=\phi_h^{gas}(\tau)+\phi_h^{elec}(\tau)-\phi_c^{elec}(\tau)-S\alpha'\left(T_I^j(\tau)-T_W^j(\tau,0)\right),\ T_I^j(0)=T_I^0$$

$$\frac{\partial T_W^j}{\partial \tau}=\beta\frac{\partial^2 T_W^j}{\partial x^2},\ \alpha'\left(T_I^j(\tau)-T_W^j(\tau,0)\right)+k\left.\frac{\partial T_W^j}{\partial \tau}\right|_{(\tau,0)}=0,\ \alpha''\left(T_W^j(\tau,L)-T_A^j(\tau)\right)+k\left.\frac{\partial T_W^j}{\partial \tau}\right|_{(\tau,L)}=0$$

$$T_W^j(0,x)=T_W^0(x),\tau\in[t,t+T]$$

$$T_I^{min}\le T_I^j(\tau)\le T_I^{max},\ j\in\{1,\ldots,S\}$$



Zavala & al. 2009

Creating the weather uncertainty

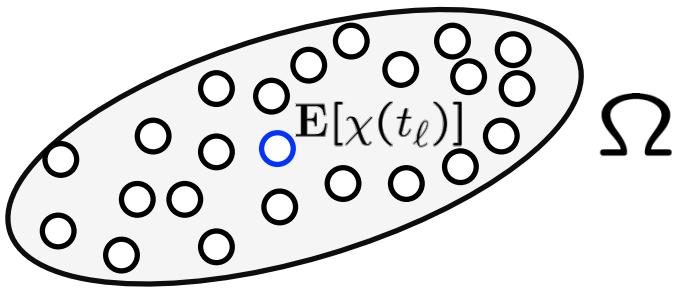
Ensemble Forecast Approach – Use WRF (Weather Research Forecast) as Black-Box

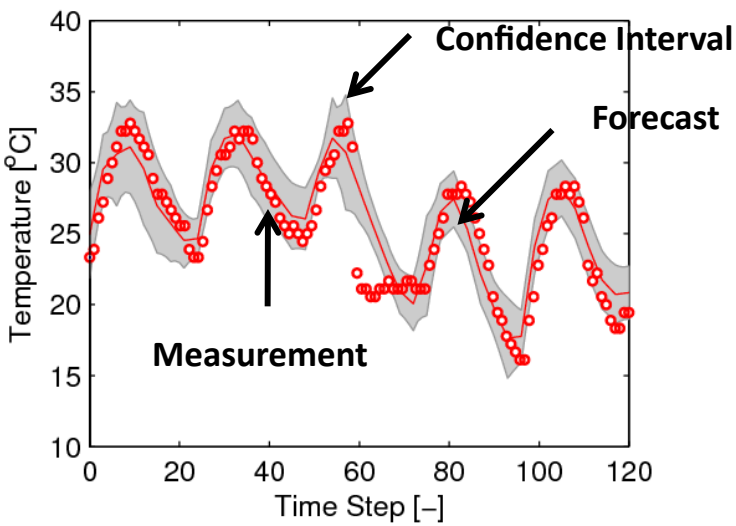
Sample Prior and Propagate Samples of Posterior Through Model

$$\mathbf{Y}_{[i,j]} := \chi_i(t_{\ell+j}) = \underbrace{\mathcal{M}(\mathcal{M}(\ldots\mathcal{M}}_{j \text{ times}}(\chi_i(t_\ell))))$$

$$\mathbf{E}[\mathbf{Y}] \approx \bar{\mathbf{Y}} := \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{Y}_{[i,:]}$$

$$\mathbf{V} \approx \frac{1}{NS-1}\sum_{i=1}^{NS} (\mathbf{Y}_{[i,:]} - \bar{\mathbf{Y}})(\mathbf{Y}_{[i,:]} - \bar{\mathbf{Y}})^T$$





Hours (August 1st-5th)

Validation Results, Pittsburgh Area 2006

5 Day Forecast and +/- 3s Intervals

Building applications

Stochastic Unit Commitment with Wind Power (SAA)

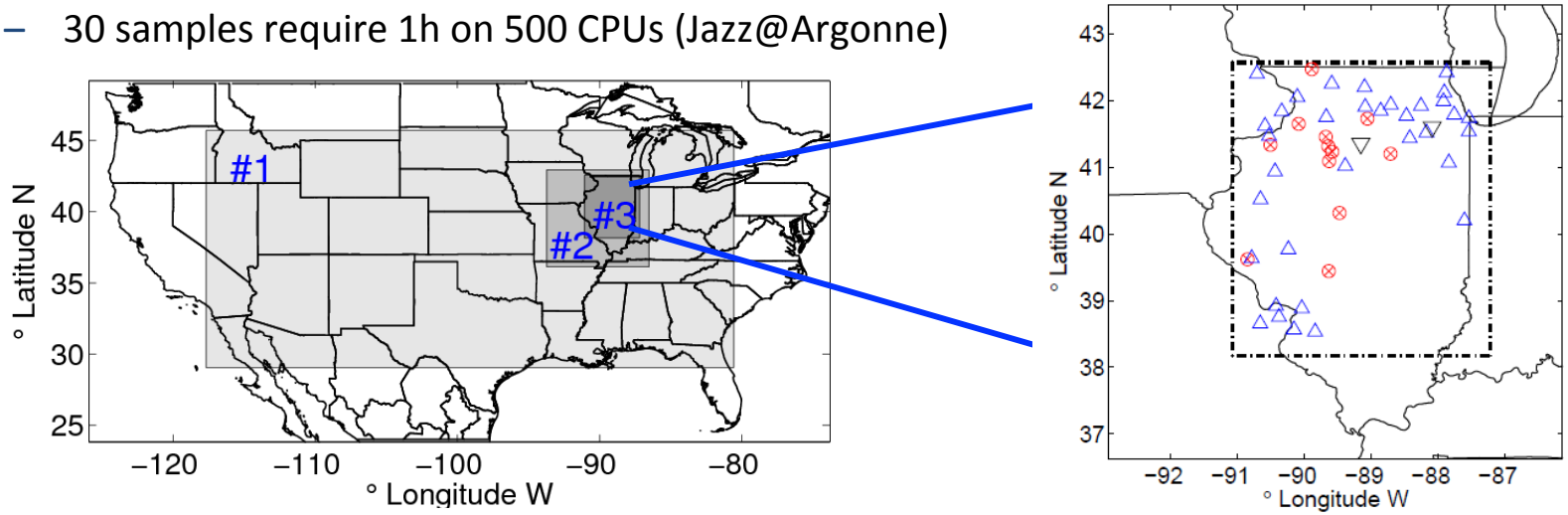
$$\min \quad \text{COST} = \frac{1}{N_s}\sum_{s\in\mathcal{S}}\left(\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{T}} c^p_{sjk} + c^u_{jk} + c^d_{jk}\right)$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}} p_{sjk} + \sum_{j\in\mathcal{N}_{wind}} p^{wind}_{sjk} = D_k, s\in\mathcal{S}, k\in\mathcal{T}$$

$$\sum_{j\in\mathcal{N}} \overline{p}_{sjk} + \sum_{j\in\mathcal{N}_{wind}} p^{wind}_{sjk} \geq D_k + R_k, s\in\mathcal{S}, k\in\mathcal{T}$$

ramping constr., min. up/down constr.

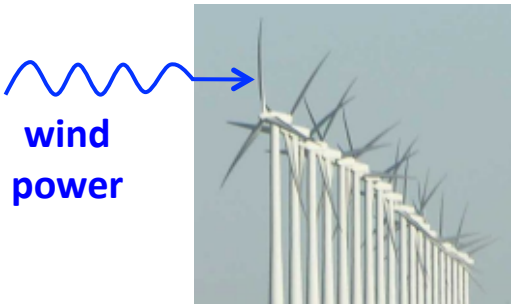
- Wind Forecast – WRF(Weather Research and Forecasting) Model
  - Real-time grid-nested 24h simulation
  - 30 samples require 1h on 500 CPUs (Jazz@Argonne)



Zavala & al 2010.

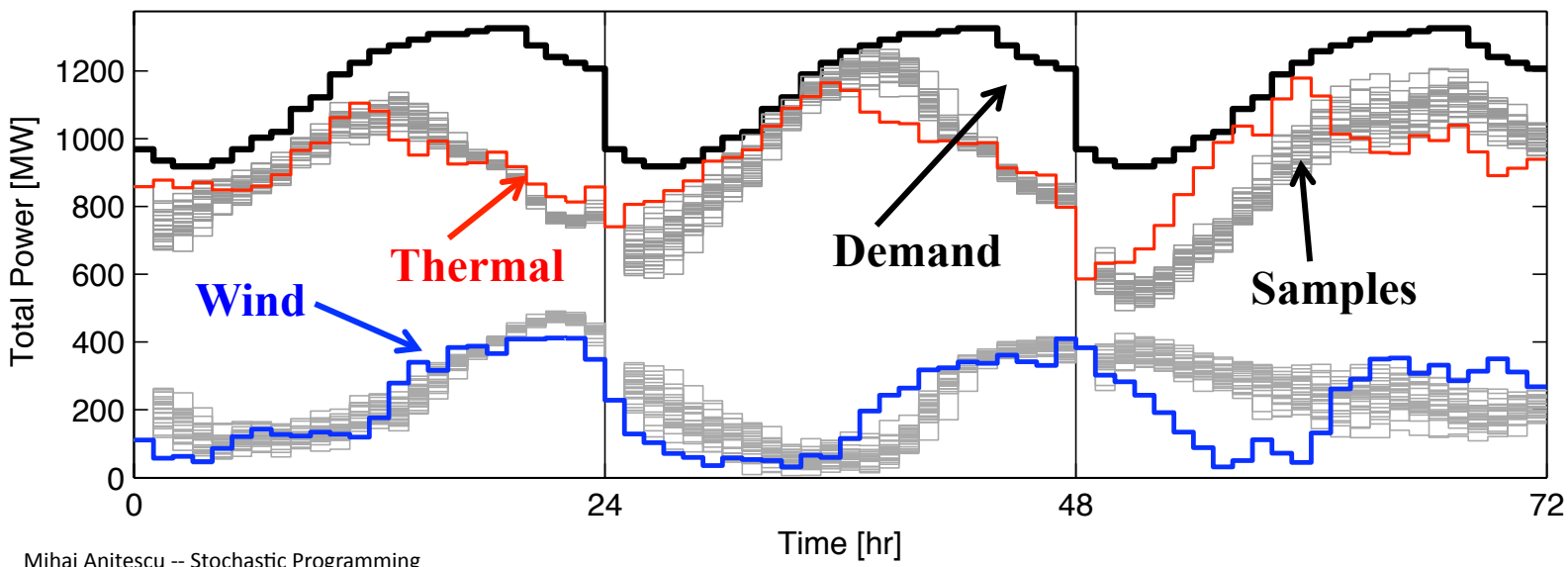
Wind power forecast and stochastic programming

- Unit commitment & energy dispatch with uncertain wind power generation for the State of Illinois, assuming 20% wind power penetration, using the same windfarm sites as the one existing today.



- Full integration with 10 thermal units to meet demands. Consider dynamics of start-up, shutdown, set-point changes

- The solution is only 1% more expensive then the one with exact information. Solution on average infeasible at 10%.



Managament under uncertainty paradigm: stochastic programming.

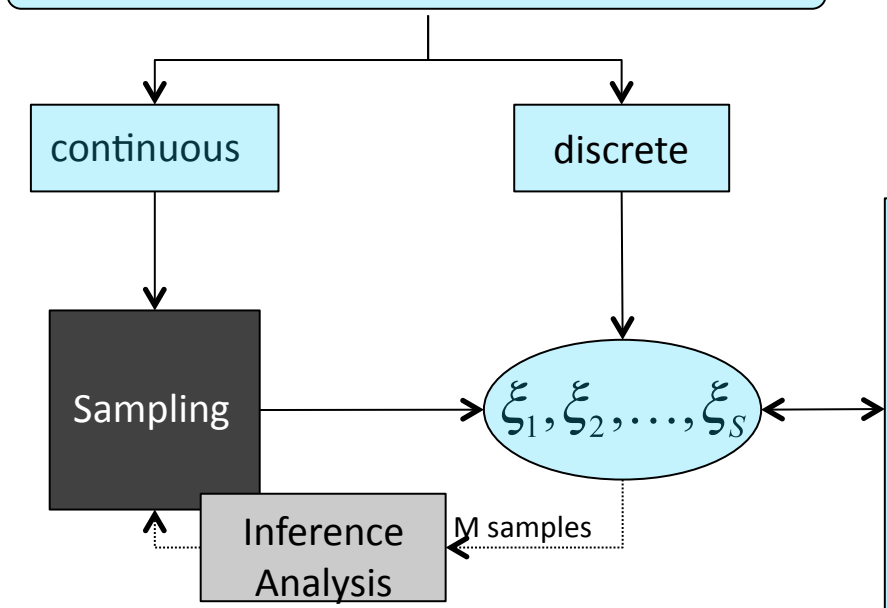
- Two-stage stochastic programming with recourse ("here-and-now")

$$\underset{x_0}{Min}\left\{ f_0(x_0) + \mathbb{E}\left[\underset{x}{Min}\, f(x,\omega) \right] \right\}$$

subj. to.

$$\begin{aligned} A_0 x_0 & & = & \quad b_0 \\ A(\omega)x_0 + & \quad B(\omega)x & = & \quad b(\omega) \\ x_0 \geq 0, & \quad x(\omega) \geq 0 & & \end{aligned}$$

- $\xi(\omega) := (A(\omega), B(\omega), b(\omega), Q(\omega), c(\omega))$



Sample average approximation (SAA)

$$\underset{x_0, x_1, x_2, \ldots, x_S}{Min} \quad f_0(x) + \frac{1}{S}\sum_{i=1}^{S} f_i(x_i)$$

subj. to.

$$\begin{aligned} A_0 x_0 & & = & \quad b_0 \\ A_k x_0 + & \quad B_k x_k & = & \quad b_k, \\ x_0 \geq 0, & \quad x_k \geq 0, & & \quad k = 1, \ldots, S \end{aligned}$$

INTERIOR POINT METHODS FOR STOCHASTIC PROGRAMMING

Linear Algebra of Primal-Dual Interior-Point Methods

Convex quadratic problem

$$\text{Min } \frac{1}{2} x^T Q x + c^T x$$

$$\text{subj. to. } Ax = b$$

$$x \geq 0$$

IPM Linear System

$$\begin{bmatrix} Q+\Lambda & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = rhs$$

Multi-stage SP

Two-stage SP

nested arrow-shaped linear system (via a permutation)

$$\begin{bmatrix}
\tilde{H}_1 & B_1^T & & & & & & 0 & 0 \\
B_1 & 0 & & & & & & A_1 & 0 \\
 & & \tilde{H}_2 & B_2^T & & & & 0 & 0 \\
 & & B_2 & 0 & & & & A_2 & 0 \\
 & & & & \ddots & & & \vdots & \vdots \\
 & & & & & \tilde{H}_S & B_S^T & 0 & 0 \\
 & & & & & B_S & 0 & A_S & 0 \\
0 & A_1^T & 0 & A_2^T & \dots & 0 & A_S^T & \tilde{H}_0 & A_0^T \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & A_0 & 0
\end{bmatrix}$$

The Direct Schur Complement Method (DSC)

- Uses the arrow shape of H

$$\begin{bmatrix} H_1 & & & & G_1^T \\ & H_2 & & & G_2^T \\ & & \ddots & & \vdots \\ & & & H_S & G_S^T \\ G_1 & G_2 & \dots & G_S & H_0 \end{bmatrix} = \begin{bmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_S & \\ L_{10} & L_{20} & \dots & L_{S0} & L_c \end{bmatrix} \begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & \ddots & & \\ & & & D_N & \\ & & & & D_c \end{bmatrix} \begin{bmatrix} L_1^T & & & & L_{10}^T \\ & L^T{}_2 & & & L_{20}^T \\ & & \ddots & & \vdots \\ & & & L^T{}_S & L_{S0}^T \\ & & & & L^T{}_c \end{bmatrix}$$

- Solving Hz=r

$$L_i D_i L_i^T = H_i, \quad L_{i0} = G_i L_i^{-T} D_i^{-1}, \quad i = 1, \dots S,$$

$$C = H_0 - \sum_{i=1}^{S} G_i H_i^{-1} G_i^T, \quad \boxed{L_c D_c L_c^T = C.}$$

Implicit factorization

$$w_i = L_i^{-1} r_i, \quad i = 1, \dots S,$$

$$\boxed{w_0 = L_c^{-1}} \left(r_0 - \sum_{i=1}^{S} L_{i0} w_i \right)$$

Back substitution

$$v_i = D_i^{-1} w_i, \quad i = 0, \dots, S$$

Diagonal solve

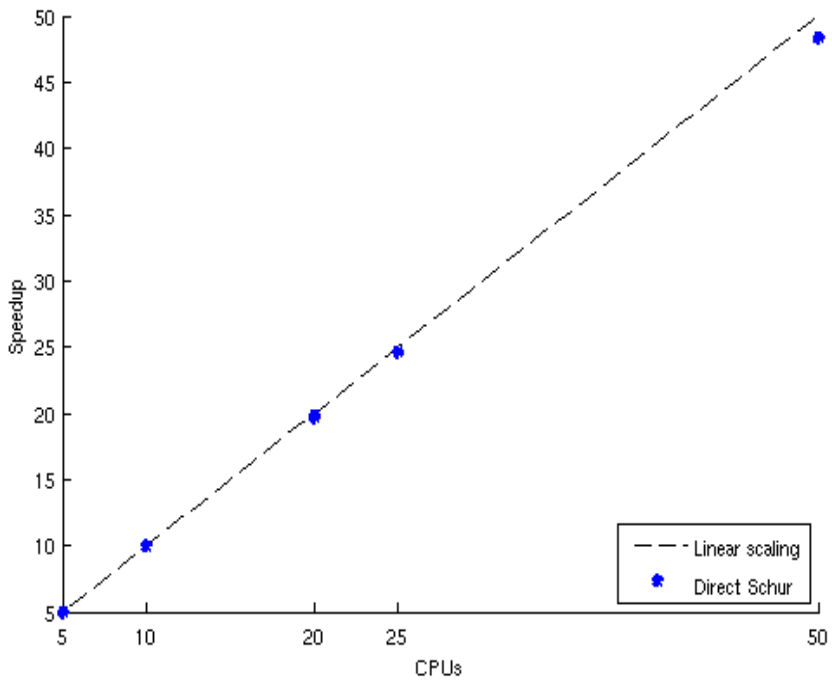
$$\boxed{z_0 = L_c^{-1} v_0}$$

$$z_i = L_i^{-T} \left(v_i - L_{i0}^T z_0 \right), \quad i = 1, \dots S.$$

Forward substitution

High performance computing with DSC

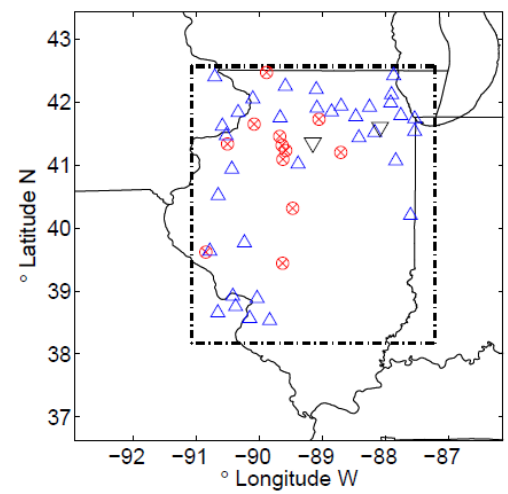
- Gondzio (OOPS) 6-stages 1 **billion** variables
- Zavala et.al., 2007 (in IPOPT)
- Our experiments (PIPS) – **strong** scaling is investigated
  - Building energy system
    - Almost linear scaling



on Fusion @ Argonne

- Unit commitment
  - Relaxation solved
  - Largest instance

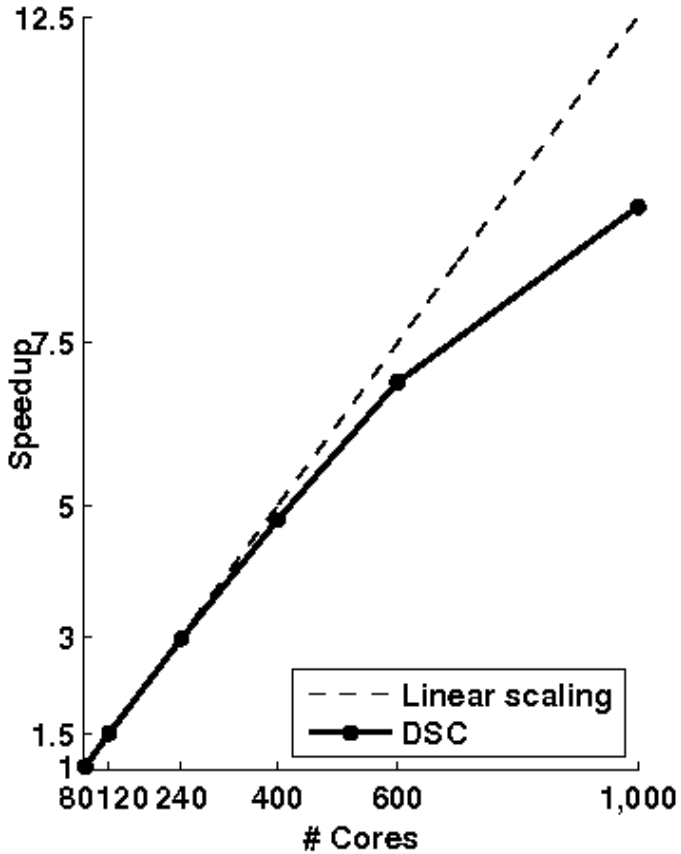
12 x



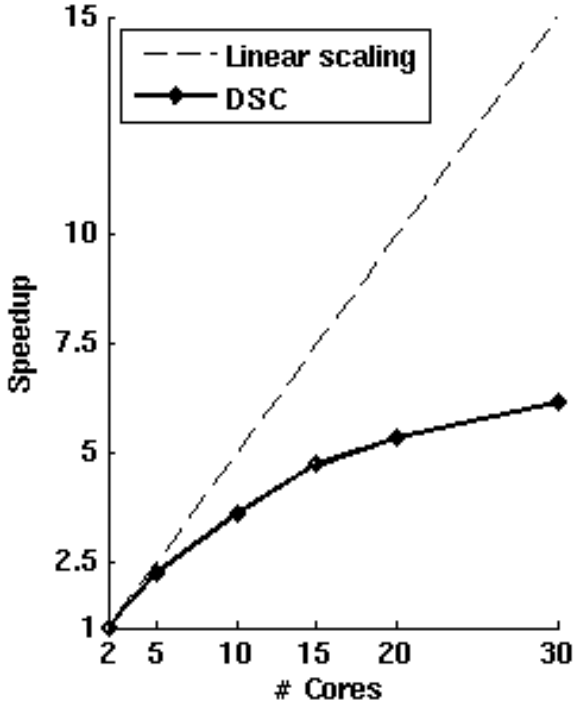
- 28.9 millions variables
- **1000 cores**

Scalability of DSC

Unit commitment
76.7% efficiency …



…but not always the case, since first stage calculations can keep everyone blocked



Large number of 1st stage variables: **38.6%** efficiency

on Fusion @ Argonne

BOTTLENECK SOLUTION 1: STOCHASTIC PRECONDITIONER

Preconditioned Schur Complement (PSC)

$$L_i D_i L_i^T = H_i, \; L_{i0} = G_i L_i^{-T} D_i^{-1}, \quad i = 1, \ldots N,$$
$$C = H_0 - \sum_{i=1}^{N} G_i H_i^{-1} G_i^T$$

$$L_M D_M L_M^T = M$$

(separate process)

$$w_i = L_i^{-1} r_i$$
$$\tilde{r}_0 = \left(r_0 - \sum_{i=1}^{N} L_{Ni} w_i \right)$$

$$v_i = D_i^{-1} w_i, \; i = 0, \ldots, N$$

$$z_0 = \mathrm{Krylov}(C, M, \tilde{r}_0)$$

$$z_i = L_i^{-T} \left(v_i - L_{i0}^T z_0 \right)$$

The Stochastic Preconditioner

- The exact structure of C is

$$C = \begin{bmatrix} \overbrace{\widetilde{Q}_0 + \frac{1}{S}\sum_{i=1}^{S}\left[A_i^T\left(B_i\widetilde{Q}_i^{-1}B_i^T\right)^{-1}A_i\right]}^{S_S} & A_0^T \\ A_0 & 0 \end{bmatrix}.$$

- IID subset of n scenarios: $\mathcal{K} = \{k_1, k_2, \ldots, k_n\}$

- The **stochastic preconditioner** (Petra & Anitescu, 2010)

$$S_n = \widetilde{Q}_0 + \frac{1}{n}\sum_{i=1}^{n}\left[A_{k_i}^T\left(B_{k_i}\widetilde{Q}_{k_i}^{-1}B_{k_i}^T\right)^{-1}A_{k_i}\right].$$

- For C use the constraint preconditioner (Keller *et. al.*, 2000)

$$M = \begin{bmatrix} S_n & A_0^T \\ A_0 & 0 \end{bmatrix}.$$

Quality of the Stochastic Preconditioner

$$S_n = \widetilde{Q}_0 + \frac{1}{n}\sum_{i=1}^{n}\left[A_{k_i}^T \left(B_{k_i} \widetilde{Q}_{k_i}^{-1} B_{k_i}^T \right)^{-1} A_{k_i} \right] \qquad S_S = \widetilde{Q}_0 + \frac{1}{S}\sum_{i=1}^{S}\left[A_i^T \left(B_i \widetilde{Q}_i^{-1} B_i^T \right)^{-1} A_i \right]$$

- **"Exponentially" better preconditioning** (Petra & Anitescu 2010)

$$\Pr(|\lambda(S_n^{-1} S_S) - 1| \geq \varepsilon) \leq 2p^4 \exp\left(-\frac{n\varepsilon^2}{2p^4 L^2 \| S_S \|_{max}^2} \right)$$

- **Proof**: Hoeffding inequality (p is dim on S; L is a bound on data)

- Assumptions on the problem's random data
  1. Boundedness
  2. Uniform full rank of $A(\omega)$ and $B(\omega)$

not restrictive (=> L)

Quality of the Constraint Preconditioner

$$M = \begin{bmatrix} S_n & A_0^T \\ A_0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} S_S & A_0^T \\ A_0 & 0 \end{bmatrix}$$

$$\underbrace{}_{p} \quad \underbrace{}_{r}$$

- $M^{-1}C$ has an eigenvalue 1 with order of multiplicity $2r$.
- The rest of the eigenvalues satisfy

$$0 < \lambda_{min}(S_n^{-1}S_S) \le \lambda(M^{-1}C) \le \lambda_{\max}(S_n^{-1}S_S).$$

- Proof: based on Bergamaschi *et. al.*, 2004.

The Krylov Methods Used for $Cz_0 = r_0$

$$\begin{bmatrix} S_S & A_0^T \\ A_0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} r_0^1 \\ r_0^2 \end{bmatrix}$$

- **BiCGStab** using constraint preconditioner M

- **Preconditioned Projected CG (PPCG)** (Gould *et. al.,* 2001)
  - Preconditioned projection onto the $KerA_0$.

$$P = Z_0 \left(Z_0^T S_n Z_0 \right)^{-1} Z_0^T$$

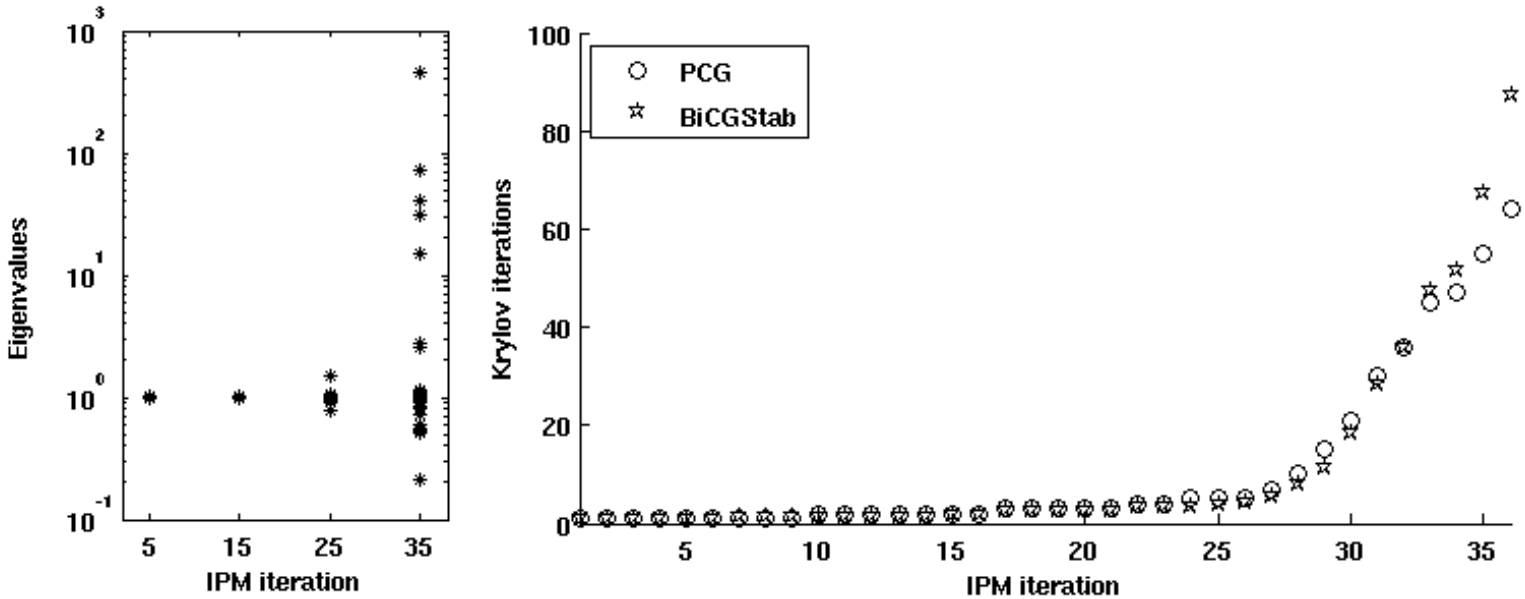
  - Does not compute the basis Z_0 for $KerA_0$. Instead,

$$g = \Pr \text{ is computed from } \begin{bmatrix} S_n & A_0^T \\ A_0 & 0 \end{bmatrix} \begin{bmatrix} g \\ u \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

  - $y_0 = (A_0 A_0^T)^{-1} A_0 \left(r_0^1 - S_N x_0 \right)$

Performance of the preconditioner

- Eigenvalues clustering & Krylov iterations



- Affected by the well-known ill-conditioning of IPMs.

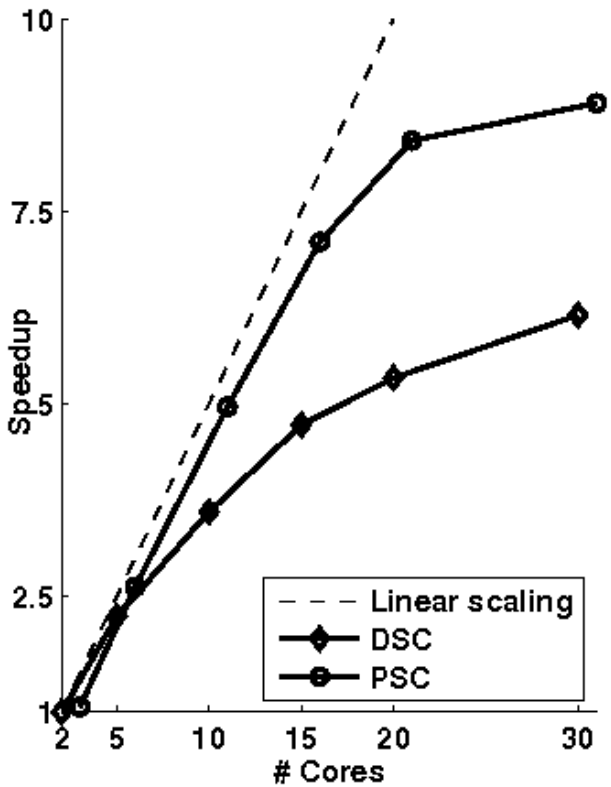
$$S_n \approx S_N \text{ and } \mathrm{S}_N \approx \mathbb{E}\left[S(\omega)\right], \text{ where}$$

$$S(\omega) = \left(Q_0 + D_0\right) + \left[A^T(\omega)\left(B(\omega)\left(Q(\omega) + D(\omega)\right)^{-1} B^T(\omega)\right)^{-1} A(\omega)\right]$$

The "Ugly" Unit Commitment Problem; PSC gets further

- DSC on P processes vs PSC on P+1 process

Optimal use of PSC – **linear scaling**



- 120 scenarios - # cores used for preconditioner
- Conclusion: PSC hides the latency well, but it eventually hits a memory wall as well.

Factorization of the preconditioner can not be hidden anymore; we need to accelerate it as well; cannot solve larger problems where improvement would likely be larger

SOLUTION 2: PARALELLIZATION OF STAGE 1 LINEAR ALGEBRA

Parallelizing the 1st stage linear algebra

- We distribute the 1st stage Schur complement system.

$$C = \begin{bmatrix} \tilde{Q} & A_0^T \\ A_0 & 0 \end{bmatrix}, \quad \tilde{Q} \text{ dense symm. pos. def.}, \quad A_0 \text{ sparse full rank.}$$

- C is treated as dense.

- Alternative to PSC for problems with large number of 1st stage variables.

- Removes the memory bottleneck of PSC and DSC.

- We investigated ScaLapack, Elemental (successor of PLAPACK)
  - None have a solver for symmetric indefinite matrices (Bunch-Kaufman);
  - LU or Cholesky only.
  - So we had to think of modifying either.

Cholesky-based LDL^T-like factorization

- $$\begin{bmatrix} \tilde{Q} & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} L & 0 \\ AL^{-T} & \bar{L} \end{bmatrix} \begin{bmatrix} I & \\ & -I \end{bmatrix} \begin{bmatrix} L^T & L^{-1}A^T \\ 0 & \bar{L}^T \end{bmatrix}, \text{ where } LL^T = \tilde{Q},\ \bar{L}\bar{L}^T = A\tilde{Q}^{-1}A^T$$
- Can be viewed as an "implicit" normal equations approach.
- In-place implementation inside Elemental: no extra memory needed.
- Idea: modify the Cholesky factorization, by changing the sign after processing p columns.
- It is much easier to do in Elemental, since this distributes elements, not blocks.
- Twice as fast as LU
- Works for more general saddle-point linear systems, *i.e.,* pos. semi-def. (2,2) block.

Distributing the 1st stage Schur complement matrix

- **All** processors contribute to **all** of the elements of the (1,1) dense block

$$\tilde{Q} = \tilde{Q}_0 + \frac{1}{S}\sum_{i=1}^{S}\left[A_i^T \left(B_i \tilde{Q}_i^{-1} B_i^T \right)^{-1} A_i \right]$$

- A large amount of inter-process communication occurs.

- Possibly more costly than the factorization itself.

- Solution: use buffer to reduce the number of messages when doing a *Reduce_scatter.*

- LDL^T approach also reduces the communication by half – only need to send lower triangle.

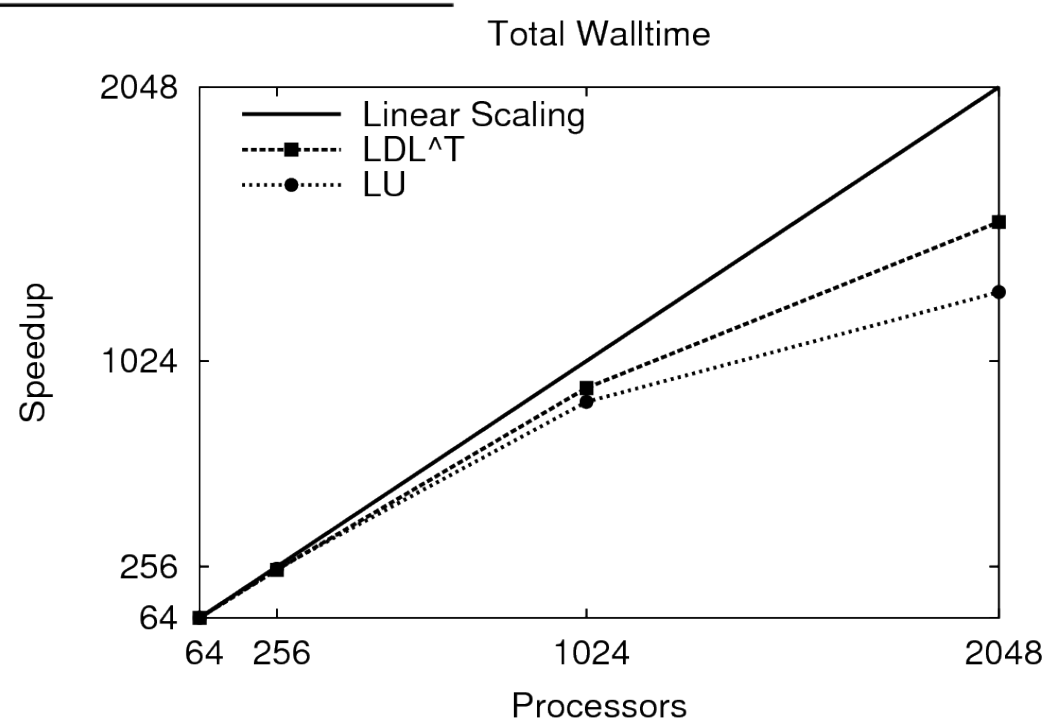
Large-scale performance

- Comparison of ScaLapack (LU), Elemental(LU), and LDL^T (1024 cores)

Units	1st Stage Size (Q+A)	Factor (Sec.) LU(S)	LU(E)	LDL^T	Reduce (Sec.) LU	LDL^T
300	23436+1224	16.59	20.04	**6.71**	54.32	**26.35**
640	49956+2584	60.67	83.24	**36.77**	256.95	**128.59**
1000	78030+4024	173.67	263.53	**90.82**	565.36	**248.22**

SAA problem:
189 million variables

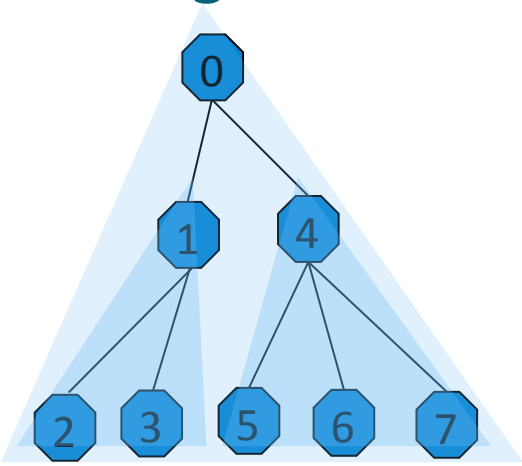
- Strong scaling
  - **90.1%** from 64 to 1024 cores;
  - 75.4% from 64 to 2048 cores.
  - > 4,000 scenarios.



A Parallel Interior-Point Solver for Stochastic Programming (PIPS)

- Convex QP SAA SP problems
- Input: users specify the scenario tree
- Object-oriented design based on OOQP
- Linear algebra: tree vectors, tree matrices, tree linear systems
- Scenario based parallelism
  - tree nodes (scenarios) are distributed across processors
  - inter-process communication based on MPI
  - dynamic load balancing
- Mehrotra predictor-corrector IPM

Multi-stage SAA SP Problems – Scenario formulation



- Depth-first traversal of the scenario tree
- **Nested** half-arrow shaped Jacobian
- Block separable obj. func.

$$Min \frac{1}{2}\sum_{i=0}^{7} x_i^T Q_i x_i + \sum_{i=0}^{7} c_i^T x_i$$

s.t.

$$\begin{array}{lllllllll}
A_0 x_0 & & & & & & & = & b_0 \\
A_1 x_0 + & B_1 x_1 & & & & & & = & b_1 \\
 & A_2 x_1 + & B_2 x_2 & & & & & = & b_2 \\
 & A_3 x_1 + & & B_3 x_3 & & & & = & b_3 \\
A_4 x_0 + & & & & B_4 x_4 & & & = & b_4 \\
 & & & & A_5 x_4 + & B_5 x_5 & & = & b_5 \\
 & & & & A_6 x_4 + & & B_6 x_6 & = & b_6 \\
 & & & & A_7 x_4 + & & & B_7 x_7 & = b_7 \\
x_0 \geq 0 & x_1 \geq 0 & x_2 \geq 0 & x_3 \geq 0 & x_4 \geq 0 & x_5 \geq 0 & x_6 \geq 0 & x_7 \geq 0 &
\end{array}$$

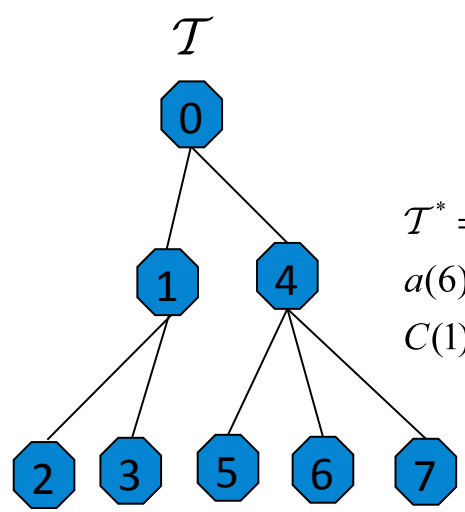
Tree Linear Algebra – Data, Operations & Linear Systems

- Data
  - Tree vector: b, c, x, etc
  - Tree symmetric matrix: Q
  - Tree general matrix: A

$$\text{Min } \frac{1}{2}x^T Q x + c^T x$$
$$\text{subj. to. } Ax = b$$
$$x \geq 0$$

- Operations

$$(u \odot v)_n = u_n \odot v_n, \forall n \in \mathcal{T}$$
$$(Qx)_n = Q_n x_n, \forall n \in \mathcal{T}$$
$$(Ax)_n = A_{a(n)} x_{a(n)} + B_n x_n, \forall n \in \mathcal{T}^*$$
$$(A^T y)_n = B_n^T y_n + \sum_{c \in C(n)} A_c^T y_c$$

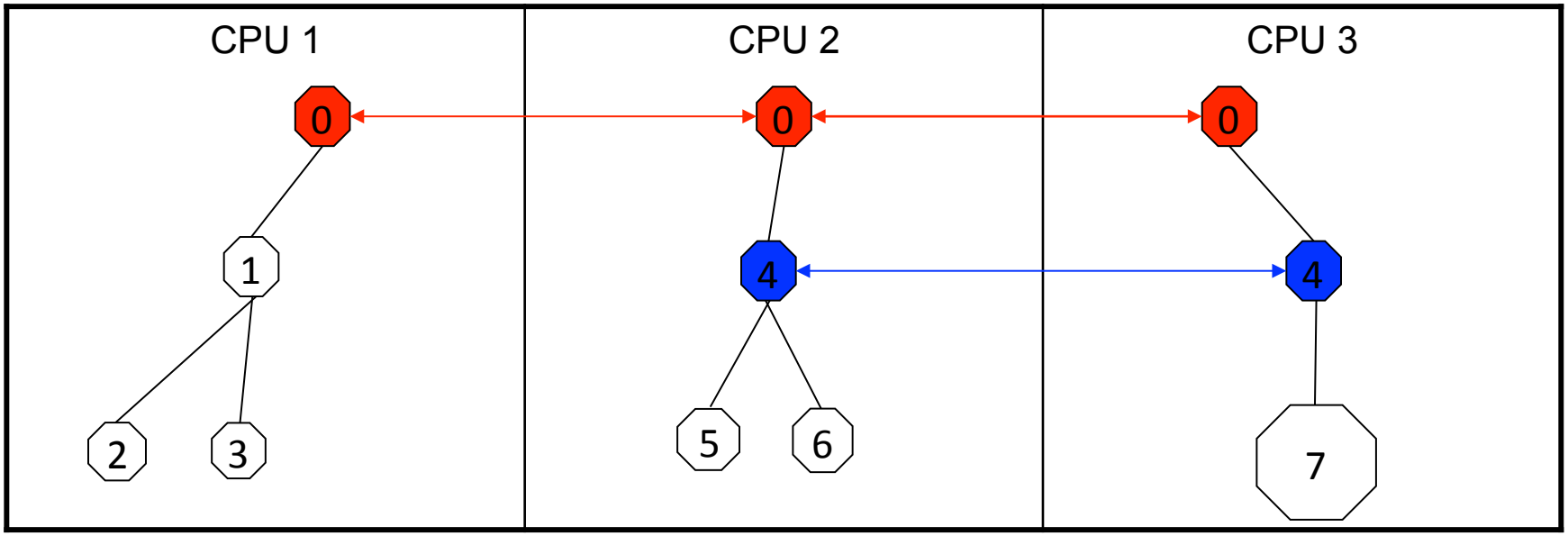


$$\mathcal{T}^* = \mathcal{T} - \{0\}$$
$$a(6) = 4 \text{ (ancestor map)}$$
$$C(1) = \{2,3\} \text{ (children map)}$$

- Linear systems: for each non-leaf node a two-stage problem is solved via Schur complement methods as previously described.

Parallelization – Tree Distribution

- The tree is distributed across processes.
- Example: 3 processes



- Dynamic load balancing of the tree
  - Number partitioning problem --> graph partitioning --> METIS

Conclusions

- The DSC method offers a good parallelism for SP in an IPM framework.
- The PSC method improves the scalability, by tackling the latency, though not the memory wall.
- Parallel direct linear algebra eliminates the memory wall; we will investigate their combination to reduce both.
- PIPS – solver for SP problems.
- PIPS is ready for larger problems: 100,000 cores.

Future work

- New math / stat
  - Asynchronous optimization
  - SAA error estimate

- New scalable methods for a more efficient software
  - Better interconnect between (iterative) linear algebra and sampling
    - importance-based preconditioning
    - multigrid decomposition
  - Target: emerging exa architectures

- PIPS
  - IPM hot-start, parallelization of the nodes
  - Ensure compatibility with other paradigms: NLP, conic progr., MILP/MINLP solvers
  - A ton of other small enhancements

- Ensure computing needs for important applications
  - Unit commitment with transmission constraints & market integration (Zavala)

References

- Victor Zavala, Emil Constantinescu, Theodore Krause, and Mihai Anitescu." On-line economic optimization of energy systems using weather forecast information." *Journal of Process Control*, Volume 19, Issue 10, 2009, Pages 1725-1736
- Emil Constantinescu, Victor Zavala, Matthew Rocklin, Sangmin Lee, and Mihai Anitescu. A Computational Framework for Uncertainty Quantification and Stochastic Optimization in Unit Commitment with Wind Power Generation. To appear in IEEE Transactions on Power Systems. Preprint ANL/MCS-P1681-1009.Electronic Version : DOI:10.1109/TPWRS.2010.2048133
- Cosmin Petra and Mihai Anitescu. "A preconditioning technique for Schur complement systems arising in stochastic optimization". Preprint ANL/MCS-P1748-0510. Submitted to *Computational Optimization and Applications*.

Thank you for your attention!

Questions?