Vectorization, communication aggregation, and reuse in stochastic and temporal dimensions

Jed Brown

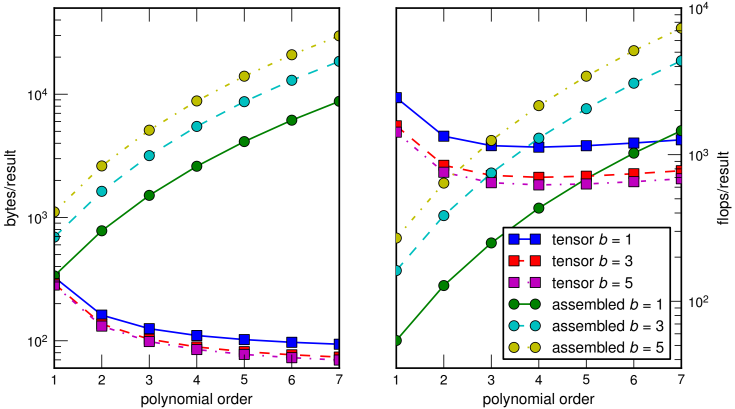
Mathematics and Computer Science Division, Argonne National Laboratory

JointLab Workshop, 2013-06-13

Premise

- Traditional abstractions are comfortable, but not optimal
- Attempts to optimize lead to “tunnel vision”
- Principles are often better applied by considering context
- Transparency in formerly-opaque abstractions enables composition that produces more effective optimization
- More regularity beyond the spatial domain

Performance of assembled versus unassembled



- High order Jacobian stored unassembled using coefficients at quadrature points, can use local AD
- Choose approximation order at run-time, independent for each field
- Precondition high order using assembled lowest order method
- Implementation $> 70\%$ of FPU peak, SpMV bandwidth wall $< 4\%$

Beyond the spatial domain: The stack is deep

- forward model: transient PDE model
  - $u(t,x;p)$ with parameters p known (initial/boundary data, coefficients, ...)
- forward model with uncertainty: stochastic representation of incompletely-modeled processes
  - $u(t,x,\mathfrak{z};p)$ with noise $\mathfrak{z}$ in stochastic space
- Design optimization with uncertainty
  - $\min_p \int_{\mathfrak{z}} f(u(t,x,\mathfrak{z},p))$
- data assimilation: infer p from sparse observations d_0 with noise $\mathfrak{y}$

$$\hat{p}(t,x,d,d_0) = \arg\min_p \int_{\mathfrak{y}} \int_{\mathfrak{z}} \|d(u(t,x,\mathfrak{z},p)+\mathfrak{y}) - d_0\|^2 + \mathrm{Prior}(p)$$

- optimal experimental design: choose "affordable" sparse observations d to minimize risk in the data assimilation problem over a region of parameter/model space

$$\hat{d} = \arg\min_d \int_p \|\hat{p}(t,x,d,d_0(p)) - p\| + \mathrm{cost}(d)$$

- "collocation" (decoupled ensemble) vs. Galerkin (coupled)

Principles

- Memory locality: cache reuse/sharing, GPU shared memory, NUMA
  - Get the maximum use out of data before we retire it from cache
- Exploitable regularity
  - Vectorization: packed SSE/AVX/QPX, avoid warp divergence
  - Coalesced loads, prefetchable streams
  - Ability to share cache between threads
  - Avoid contention in case of overlapped writes

The quest for exploitable regularity

Spatial domain

- Complex geometry
- Non-smooth transient features (e.g., fracture, corners, shocks)
- Free boundaries
- Boundary conditions
- Spatial adaptivity
- Only distribute spatial domain
- Pipeline length is costly

Temporal/stochastic domain

- Simple/no geometry
- Internally smoother (branch jumps rare)
- Little or no boundary conditions (initial conditions)
- Global time steps for stiff problems (but local for hyperbolic)
- Data size up to proportional to entire simulation

- Aggregate: vectorize, amortize communication; same total work
- Synergy: mutually-beneficial reuse/accelerated convergence

The quest for exploitable regularity

Spatial domain

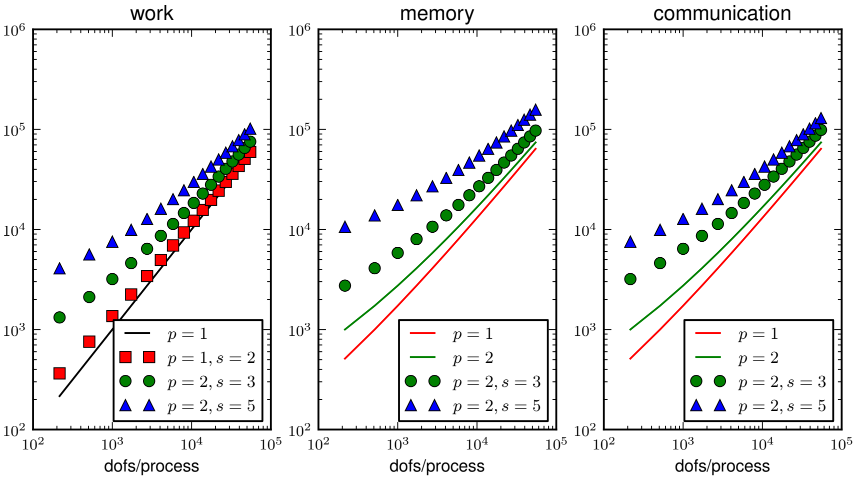
- Complex geometry
- Non-smooth transient features (e.g., fracture, corners, shocks)
- Free boundaries
- Boundary conditions
- Spatial adaptivity
- Only distribute spatial domain
- Pipeline length is costly

Temporal/stochastic domain

- Simple/no geometry
- Internally smoother (branch jumps rare)
- Little or no boundary conditions (initial conditions)
- Global time steps for stiff problems (but local for hyperbolic)
- Data size up to proportional to entire simulation

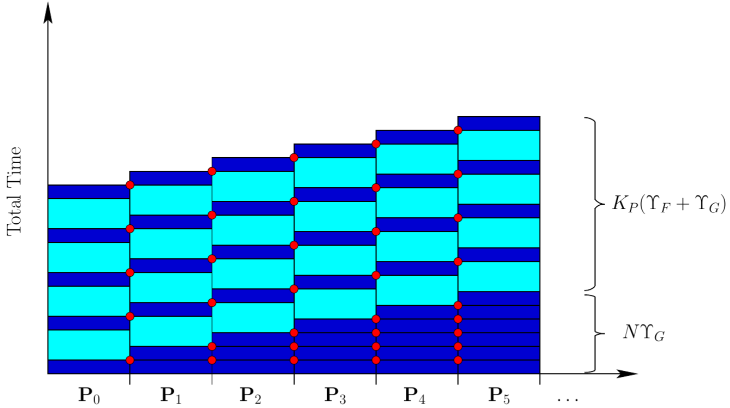
- Aggregate: vectorize, amortize communication; same total work
- Synergy: mutually-beneficial reuse/accelerated convergence

Example: s-step methods in 3D



- Amortizing message latency is most important for strong-scaling
- s-step methods have high overhead for small subdomains
- Limited choice of preconditioners (none optimal)

Example: space-time methods (multilevel SDC/Parareal)



- PFASST algorithm (Emmett and Minion, 2013)
- Zero-latency messages (cf. performance model of s-step)
- Spectral Deferred Correction: iterative, converges to IRK (Gauss, Radau, . . .)
- Stiff problems use implicit basic integrator (synchronizing on spatial communicator)

Merging Implicit Runge-Kutta with Multigrid

- Expand view to space-time Implicit RK problem
- PFASST is a line smoother (accurate solve in spatial domain)
- PFASST uses finest-possible decomposition in time (latency-intolerant)
- What about chunky space-time domains and a space-time smoother?
- Aggregate
  - Amortize or pipeline communication over stages (no overhead)
  - Vectorize nonlinear residual over stages
- Synergy
  - Reuse point/cell Jacobian in smoother for all stages (point-modified Newton, cf. Implicit RK)
  - Frozen τ for parabolic problems (Brandt and Greenwald, *Parabolic multigrid revisited*, 1991)
  - Selectively multiplicative in time (Vandewalle and Horton, *Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods*, 1995)
  - Block/recycling Krylov acceleration

Merging Implicit Runge-Kutta with Multigrid

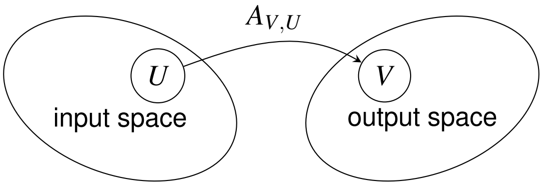
- Expand view to space-time Implicit RK problem
- PFASST is a line smoother (accurate solve in spatial domain)
- PFASST uses finest-possible decomposition in time (latency-intolerant)
- What about chunky space-time domains and a space-time smoother?
- Aggregate
  - Amortize or pipeline communication over stages (no overhead)
  - Vectorize nonlinear residual over stages
- Synergy
  - Reuse point/cell Jacobian in smoother for all stages (point-modified Newton, cf. Implicit RK)
  - Frozen τ for parabolic problems (Brandt and Greenwald, *Parabolic multigrid revisited*, 1991)
  - Selectively multiplicative in time (Vandewalle and Horton, *Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods*, 1995)
  - Block/recycling Krylov acceleration

Merging Implicit Runge-Kutta with Multigrid

- Expand view to space-time Implicit RK problem
- PFASST is a line smoother (accurate solve in spatial domain)
- PFASST uses finest-possible decomposition in time (latency-intolerant)
- What about chunky space-time domains and a space-time smoother?
- Aggregate
  - Amortize or pipeline communication over stages (no overhead)
  - Vectorize nonlinear residual over stages
- Synergy
  - Reuse point/cell Jacobian in smoother for all stages (point-modified Newton, cf. Implicit RK)
  - Frozen τ for parabolic problems (Brandt and Greenwald, *Parabolic multigrid revisited*, 1991)
  - Selectively multiplicative in time (Vandewalle and Horton, *Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods*, 1995)
  - Block/recycling Krylov acceleration

A notion of coupling rank

- "Full space" (spatial, temporal, parameters, stochastic, ...)

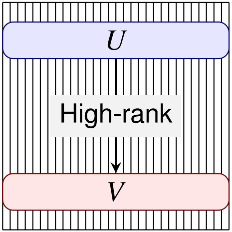


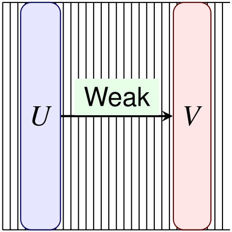
- $A_{V,U}$: dependence of solution in subdomain V on data from subdomain U
- Essential rank k of $A_{V,U}$ is number of singular values greater than chosen relative tolerance
- Crude lower bound: at least k units of information must be communicated from U to V
- Parallel distribution: high-rank couplings should be "nearby"

Implication of essential coupling rank for anisotropy

Anisotropic diffusion:

$$-\nabla \cdot (\kappa \nabla u) = f, \qquad \kappa = \begin{pmatrix} \varepsilon & 0 \\ 0 & 1 \end{pmatrix}$$





- Identical distribution for input and output spaces is natural.
- The k items of data can be communicated in different ways:
  - Increasingly-large subdomains at greater distance (tree-code)
  - Interaction through coarse grid (multigrid, fast multipole)

Essential rank in space-time

- Higher rank coupling in time than space
  - Hyperbolic equation $\Delta x = \lambda_{\max} \Delta t$: causality cone is steeper in time than space (equivalent for fastest wave).
  - Parabolic: Green's function decays faster in space than in time (depends on Δt)

Ramifications and research priorities

- No need to focus on strict independence
  - We will communicate globally anyway
  - Choose distributions to make long-distance (in space/time/stochastic) communications low-rank
  - "Treecode-to-FMM" transformations to further exploit low-rank
- Exploit structure to aggregate communication and vectorize
- Raise temporal and stochastic dimensions to first-class
  - Think of algorithms in full space, then map to space-time computational strategy
- Adaptive recognition of reusability/synergistic structure
- Load balancing due to adaptive spatio-temporal-stochastic reuse
- Evolution of software interfaces
  - Nuanced problem structure
  - Reusable components that do less than "solve" the sub-problem
- Extend analysis to "full-space" methods
- Programming tools
  - Unintrusive manipulation of logical vector length
  - Support for judicious use of cross-lane operations
  - Asynchronous and aggregated communication

- Maximize science per Watt
- Huge scope remains at problem formulation
- Raise level of abstraction at which a problem is formally specified
- Algorithmic optimality is crucial