Overview of TAMU ASCI Project

Marvin L. Adams

Texas A&M University

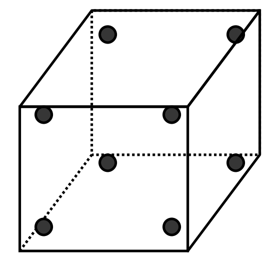
Parallel-S_N Workshop #3

May 1-2, 2001

Texas A&M University

The ASCI program needs efficient massively parallel particle transport.

- **Accelerated Strategic Computing Initiative:**
  - ⇒ ***if you can't test, you must compute!***
  - ⇒ ***focus: 3D multi-physics simulations***
- **ASCI-scale 3D multi-physics:**
  - ⇒ ***centuries of CPU time***
  - ⇒ ***Terabytes of RAM***
  - ⇒ ***not possible without efficient use of massively parallel computers!***
- **Particle transport dominates CPU and memory**
- **"Efficient use" means more than scaling**
  - ⇒ ***easy parallel algorithms may take forever to converge***
  - ⇒ ***beware of poor single-CPU performance***
  - ⇒ ***need combination of iterative convergence, single-CPU performance (cache), and parallel scaling***

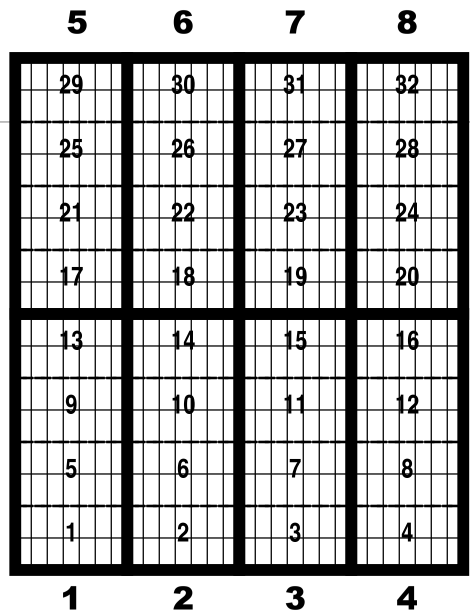


Angular intensity:
8 corners
x 50 energy groups
x 300 directions
= 120,000 unknowns
per cell per timestep

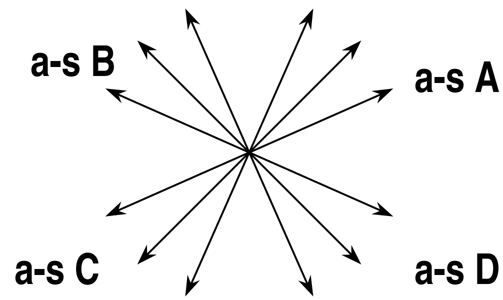
We believe we have found a powerful approach to parallel transport.

- **We try to “sweep” as much as is practical.**
  - ⇒ ***KBA showed the way; we try to go a bit farther***
  - ⇒ ***On arbitrary grids we use same philosophy***
  - ⇒ ***Implementation includes continuum from sweeps to block-Jacobi***
- **Our approach:**
  - ⇒ ***find and break cycles in grid; create DAG’s***
  - ⇒ ***“schedule” the problem***
    - choose spatial sub-domains; choose “cell-sets” in each
    - organize energy groups into “group-sets” and directions into “angle-sets”
    - for each angle-set, order the cell-sets and the cells within cell-sets
  - ⇒ ***for each group-set:***
    - set source for in-scattering from other group-sets
    - For each within-group-set scattering iteration:
      - → *get total source*
      - → ***execute in parallel** all (angle-set, cell-set) pairs in this group-set (KBA-ish)*
      - → *execute in parallel the **acceleration** step (TSA)*
      - → *test for convergence of within-group-set scattering iteration*
    - test for convergence of group-set – to – group-set scattering

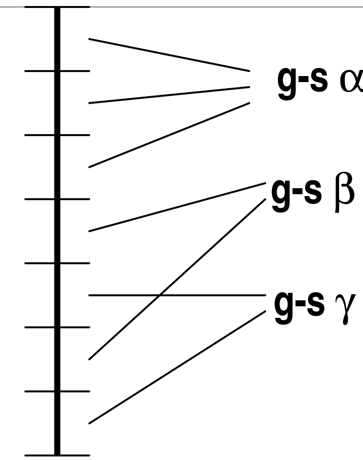
space

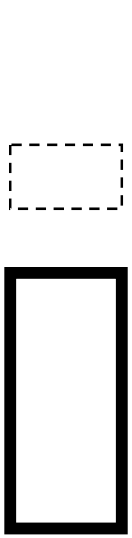


direction
(or “angle”)



energy





Example problem:

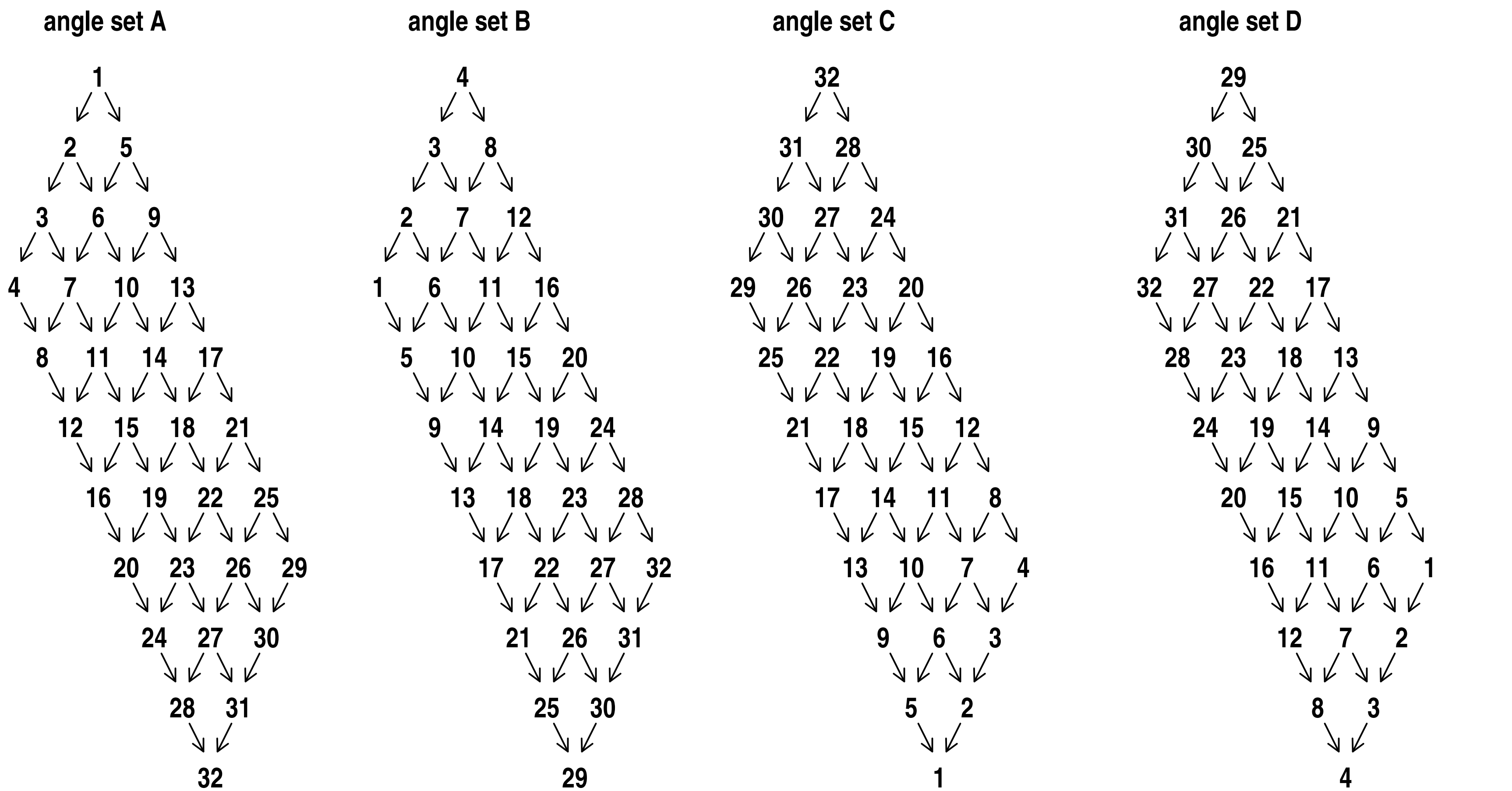
2 spatial dimensions, 8 processors, 8x4 = 32 cell-sets, 8x4x14=448 cells,

12 angles, 4 angle-sets,

7 energy groups, 3 group-sets, same directions and angle-sets for each group-set.

See next page for execution order.

For each group-set, we have the following DAGs, one for each of the four angle-sets. Each node in each DAG is a cell-set and is so numbered. Cell-set COLORS correspond to processors – there are 8 different colors used for the 8 different processors.



This approach keeps processors busy with useful work during a pure sweep.

- **16 units of work take 17 steps (93%)**
- **7 cells/cell-set changes this to 32/33 (97%)**
- **1 angle/angle-set changes it to 48/49 (98%) or 96/97 (99%)**

"step"	processor							
	1	2	3	4	5	6	7	8
1	(A,1)			(B,4)	(D,29)			(C,32)
2	(A,5)	(A,2)	(B,3)	(B,8)	(D,25)	(D,30)	(C,31)	(C,28)
3	(A,9)	(B,2)	(A,3)	(B,12)	(D,21)	(C,30)	(D,31)	(C,24)
4	(A,13)	(1,6)	(B,7)	(B,16)	(D,17)	(D,26)	(C,27)	(C,20)
5	(B,1)	(B,6)	(A,7)	(A,4)	(C,29)	(C,26)	(D,27)	(D,32)
6	(B,5)	(A,10)	(B,11)	(A,8)	(C,25)	(D,22)	(C,23)	(D,28)
7	(D,13)	(B,10)	(A,11)	(C,16)	(A,17)	(C,22)	(D,23)	(B,20)
8	(D,9)	(A,14)	(B,15)	(C,12)	(A,21)	(D,18)	(C,19)	(B,24)
9	(B,9)	(B,14)	(A,15)	(A,12)	(C,21)	(C,18)	(D,19)	(D,24)
10	(D,5)	(D,14)	(C,15)	(C,8)	(A,25)	(A,18)	(B,19)	(B,28)
11	(B,13)	(C,14)	(D,15)	(A,16)	(C,17)	(B,18)	(A,19)	(D,20)
12	(D,1)	(D,10)	(C,11)	(C,4)	(A,29)	(A,22)	(B,23)	(B,32)
13	(C,13)	(C,10)	(D,11)	(D,16)	(B,17)	(B,22)	(A,23)	(A,20)
14	(C,9)	(D,6)	(C,7)	(D,12)	(B,21)	(A,26)	(B,27)	(A,24)
15	(C,5)	(C,6)	(D,7)	(D,8)	(B,25)	(B,26)	(A,27)	(A,28)
16	(C,1)	(D,2)	(C,3)	(D,4)	(B,29)	(A,30)	(B,31)	(A,32)
17		(C,4)	(D,3)			(B,30)	(A,31)	

Our approach and flexible implementation allows optimization on many fronts.

- **Work is done in units of "chunks"**

  chunk (i,j,k) = group-set i $\times$ angle-set j $\times$ cell-set k.

- **We execute the work for one "chunk" as follows:**

```
loop over cells in cell-set (in order specified by scheduler)
    loop over angles in angle-set
        loop over groups in group-set
```

- **Tradeoffs:**
  - ***larger cell-sets $\Rightarrow$ more work per communication (fewer messages)***
  - ***smaller cell-sets $\Rightarrow$ faster pipe fill***
  - ***more groups per group-set $\Rightarrow$ better cache re-use, less indirection***
  - ***fewer groups per group-set $\Rightarrow$ fewer iterations? (prob-dependent)***
  - ***more angles per angle-set $\Rightarrow$ better cache re-use, less indirection***
  - ***fewer angles per angle-set $\Rightarrow$ truer sweeps, faster pipe fill***

- **Code structure allows:**

  $\Rightarrow$ ***1 to N_{groups} groups per group-set***

  $\Rightarrow$ ***1 to N_{octant} angles per angle-set.***

  $\Rightarrow$ ***great flexibility in cells per cell-set***

- **Optimum combination depends on architecture and physics.**

The code structure is quite flexible.