TraFiC^4 and Supercomputing

Andreas Kabel

Stanford Linear Accelerator Center

1. **Parallelization**
2. **The NERSC Facility**
3. **Parallelizing TraFiC^4**

Parallelization

DOCTOR FUN

9 Jan 2002

"This is our experimental RAIB array."

Parallelization

- Today's prevailing scheme: SPMD: Single Program, Multiple Data
- Have one code run on different machines on different sets of data
- Synchronization/data exchange checkpoints necessary
- Current trend: Commodity hardware/Free OS ('Beowulf clusters')
- Emerging standard: MPI (MPIch)
- IBM RS/6000, Sun, Beowulf-type clusters, heterogeneous networks

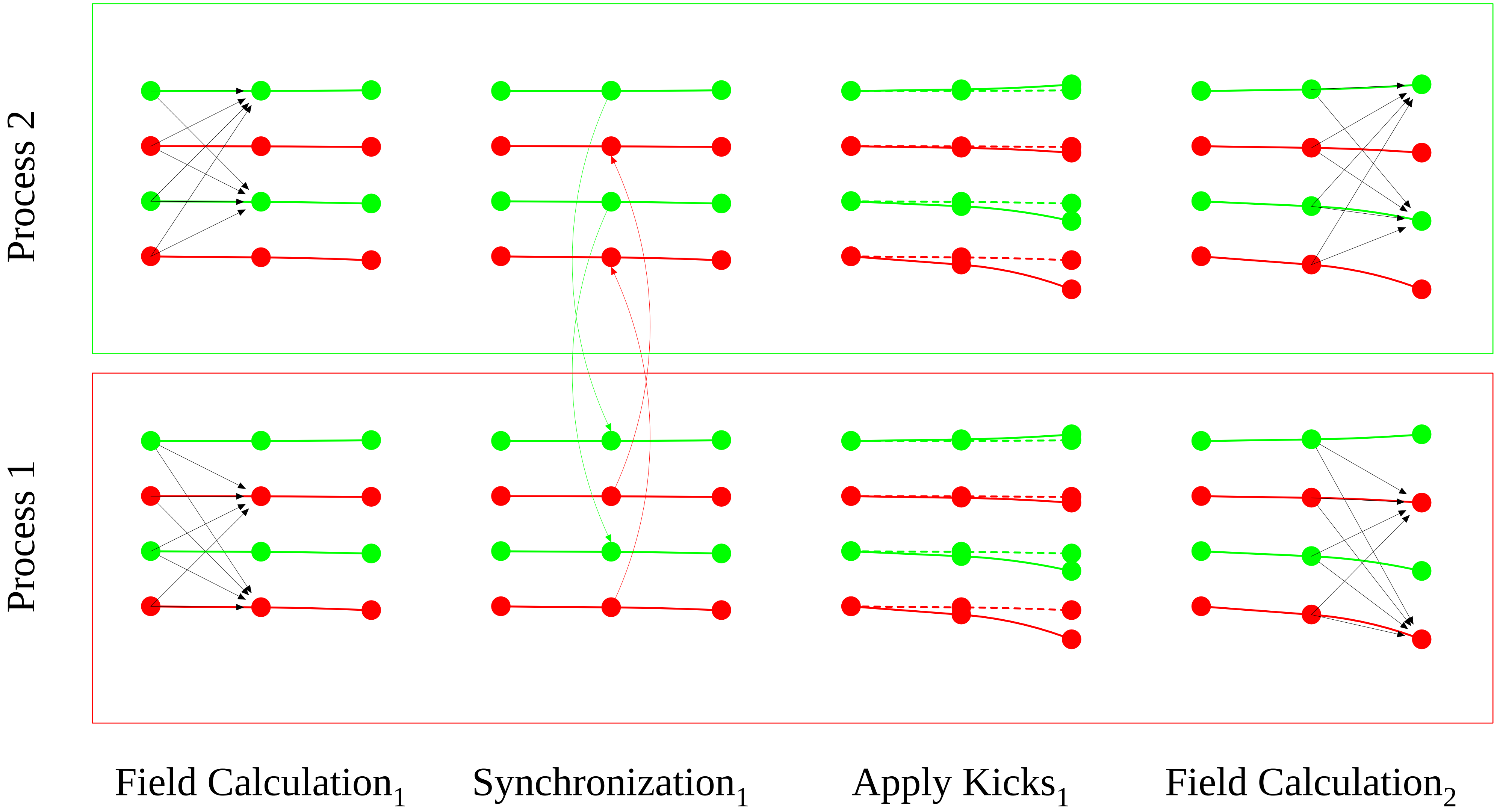
NERSC Facility

IBM SP cluster

Parallelizing TraFiC4

- Natural candidate: expensive calculation; small amounts of data
- Parallelize the Kicks algorithm by SPMD:
  - Let each proces operate on *all* particles
  - Each process calculates the kicks only for its share of particles (expensive)
  - results are broadcast in a synchronization step (cheap)
  - Kick particles with shared results, loop

Parallelizing TraFiC4



Process 2

Process 1

Field Calculation$_1$ Synchronization$_1$ Apply Kicks$_1$ Field Calculation$_2$

Load Balancing

- Works fine for sufficiently synchronized machines
- Doesn't work at all for heterogeneous clusters
- Distribute particle responsibility according to expected performance (800 MHz machine gets $2/3$, 400 MHz gets $1/3$)
- Refinement: Do reassignment dynamically: measure speed for last round, redistribute particle responsibilty accordingly
- → workable on heterogeneous clusters of commodity hardware

Experiences

- Implemented on NERSC, using MPI
- Almost linear behavior
- We were able to burn 10 years of CPU time in a fortnight
- Shielded calculations for CLIC experiment series could be completed (→ Experimental talk)
- Also works well in networked cluster of Linux machines (T. Limberg, Ph. Piot)