***N* Tropy**: A Framework for Parallel Data Analysis

Harnessing the Power of Parallel Grid Resources for Astronomical Data Analysis

Jeffrey P. Gardner,
Andy Connolly

Pittsburgh Supercomputing Center
University of Pittsburgh
Carnegie Mellon University

PITTSBURGH SUPERCOMPUTING CENTER

TERAGRID

Mining the Universe can be Computationally Expensive

- Astronomy now generates ~ 1TB data per night
- With VOs, one can pool data from multiple catalogs.
- Computational requirements are becoming much more extreme relative to current state of the art.
- There will be many problems that would be impossible without parallel machines.
- Example: N-Point correlation functions for the SDSS
  - 2-pt: CPU hours
  - 3-pt: CPU weeks
  - 4-pt: 100 CPU years!
- There will be many more problems for which throughput can be substantially enhanced by parallel machines.

PITTSBURGH SUPERCOMPUTING CENTER

Types of Parallelism

- Data Parallel (or “Embarrassingly Parallel”):
  - Example:
    - 100,000 QSO spectra
    - Each spectrum takes ~1 hour to reduce
    - Each spectrum is computationally independent from the others
  - If you have root access to a Grid resource:
    - Solution for “traditional” enviroment: Condor
    - VOs will provide a integrated workflow solution (e.g. Pegasus)
  - Running on shared resources like the TeraGrid is more difficult
    - TeraGrid has no metascheduler
    - TeraGrid batch systems cannot handle 100,000 independent work units
    - Solution: GridShell (talk to me if you are interested!)

PITTSBURGH SUPERCOMPUTING CENTER

Types of Parallelism

- Tightly Coupled Parallelism (What this talk is about):
  - Data and computational domains overlap
  - Examples:
    - N-Point correlation functions
    - New object classification
    - Density estimation
    - Intersections in parameter space
  - Solution(?):
    - *N* Tropy

PITTSBURGH SUPERCOMPUTING CENTER

The Challenge of Parallel Data Analysis

- Parallel programs are hard to write!
  - Steep learning curve to learn parallel programming
  - Lengthy development time
- Parallel world is dominated by simulations:
  - Code is often reused for many years by many people
  - Therefore, you can afford to spend lots of time writing the code.
- Data Analysis does not work this way:
  - Rapidly changing scientific inqueries
  - Less code reuse
- Data Analysis requires rapid software development!
- Even the simulation community rarely does data analysis in parallel.

PITTSBURGH SUPERCOMPUTING CENTER

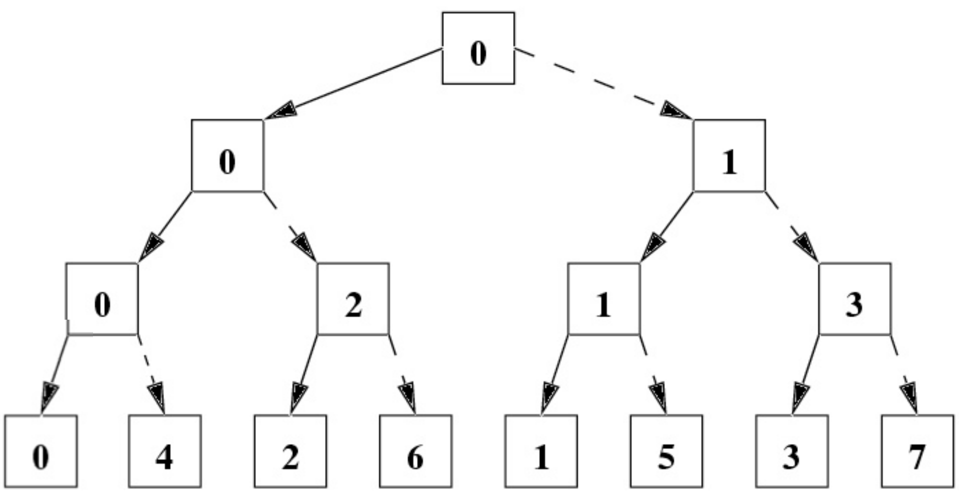
The Goal

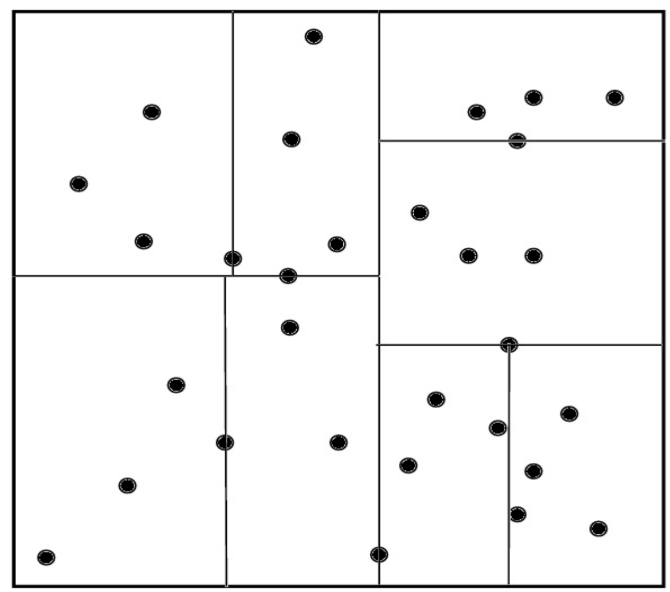
- GOAL: Minimize development time for parallel applications.
- GOAL: Allow scientists who don't have the time to learn how to write parallel programs to still implement their algorithms in parallel.
- GOAL: Provide seamless scalability from single processor machines to TeraGrid platforms
- GOAL: Do not restrict inquiry space.

PITTSBURGH SUPERCOMPUTING CENTER

Methodology

- Limited Data Structures:
  - Most (all?) efficient data analysis methods use grids or trees.
- Limited Methods:
  - Analysis methods perform a limited number of operations on these data structures.



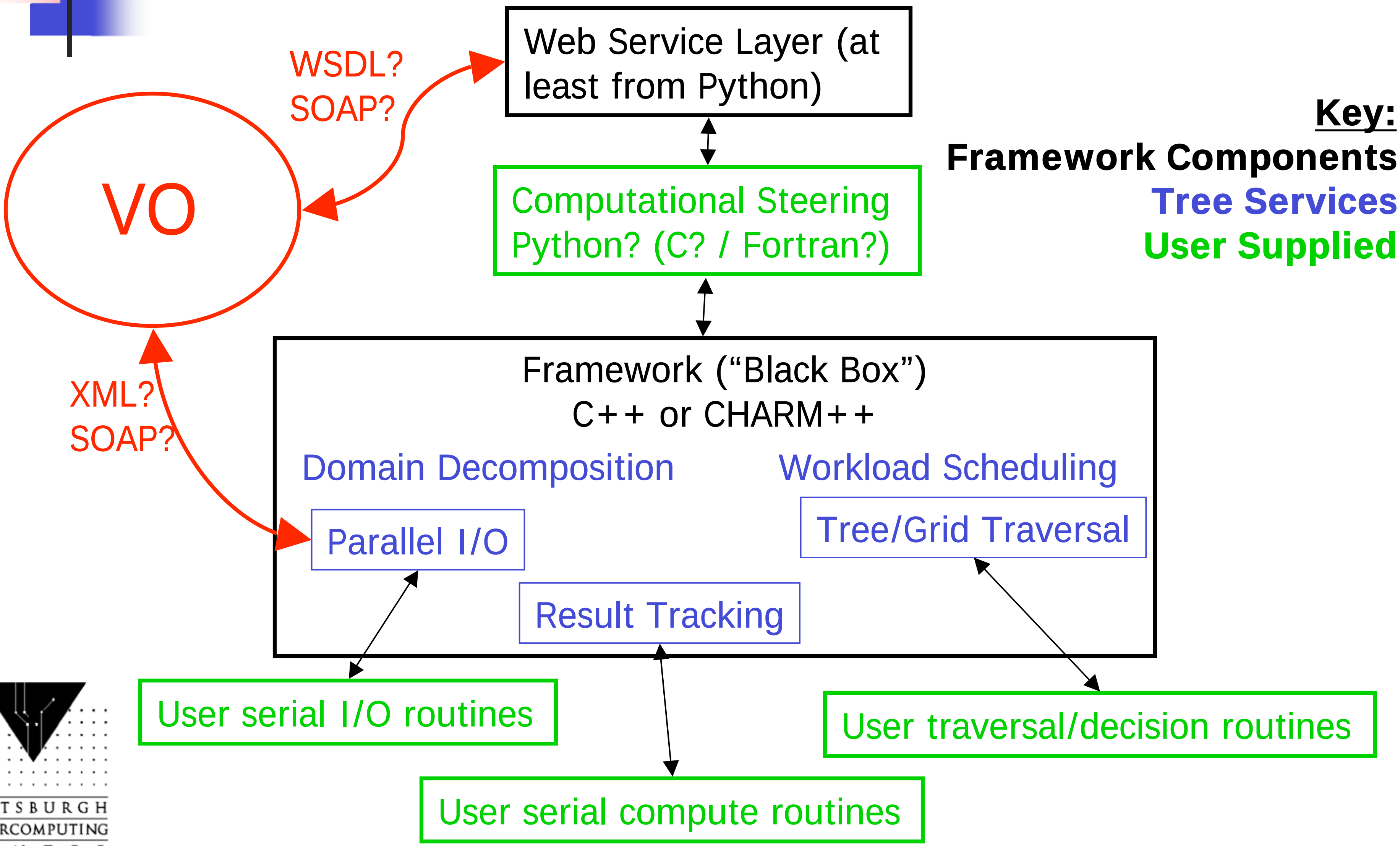


Methodology

- **Examples**:
  - Fast Fourier Transform
    - Abstraction: Grid
    - Method: Global Reduction
  - N-Body Gravity Calculation
    - Abstraction: Tree
    - Method: Global Top-Down TreeWalk
  - 2-Point Correlation Function Calculation
    - Abstraction: Tree
    - Method: Global Top-Down TreeWalk

PITTSBURGH
SUPERCOMPUTING
CENTER

Vision: A Parallel Framework



PITTSBURGH
SUPERCOMPUTING
CENTER

Proof of Concept: PHASE 1
(complete)

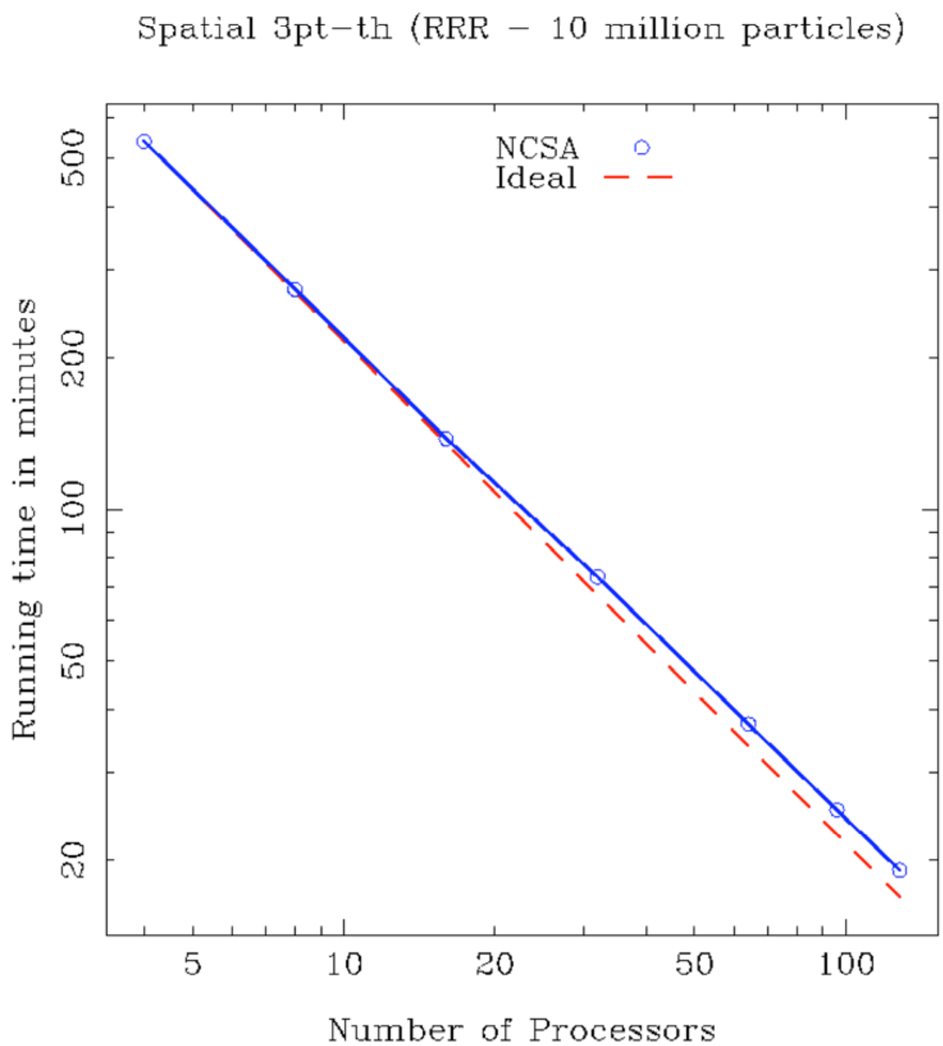
- Convert parallel N-Body code "PKDGRAV*" to 3-point correlation function calculator by modifying existing code as little as possible.
  - *PKDGRAV developed by Tom Quinn, Joachim Stadel, and others at the University of Washington
- PKDGRAV (aka GASOLINE) benefits:
  - Highly portable
    - MPI, POSIX Threads, SHMEM, Quadrics, & more
  - Highly scalable
    - 92% linear speedup on 512 processors
- Development time:
  - Writing PKDGRAV: ~10 FTE years (could be rewritten in ~2)
  - PKDGRAV -> 2-Point: 2 FTE weeks
  - 2-Point -> 3-Point: >3 FTE months

PITTSBURGH SUPERCOMPUTING CENTER

PHASE 1 Performance

10 million particles
Spatial 3-Point
3->4 Mpc

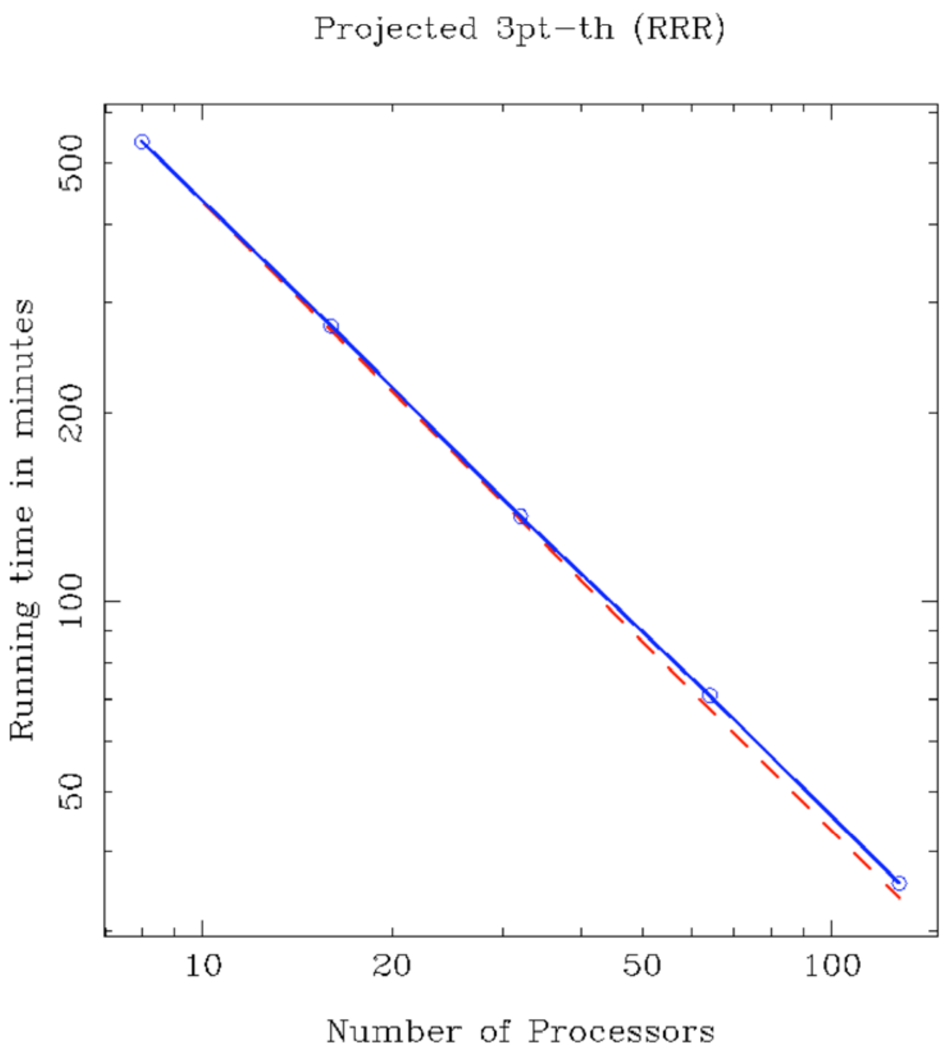
(SDSS DR1 takes less than 1 minute with perfect load balancing)



PITTSBURGH SUPERCOMPUTING CENTER

PHASE 1 Performance

10 million particles
Projected 3-Point
0->3 Mpc

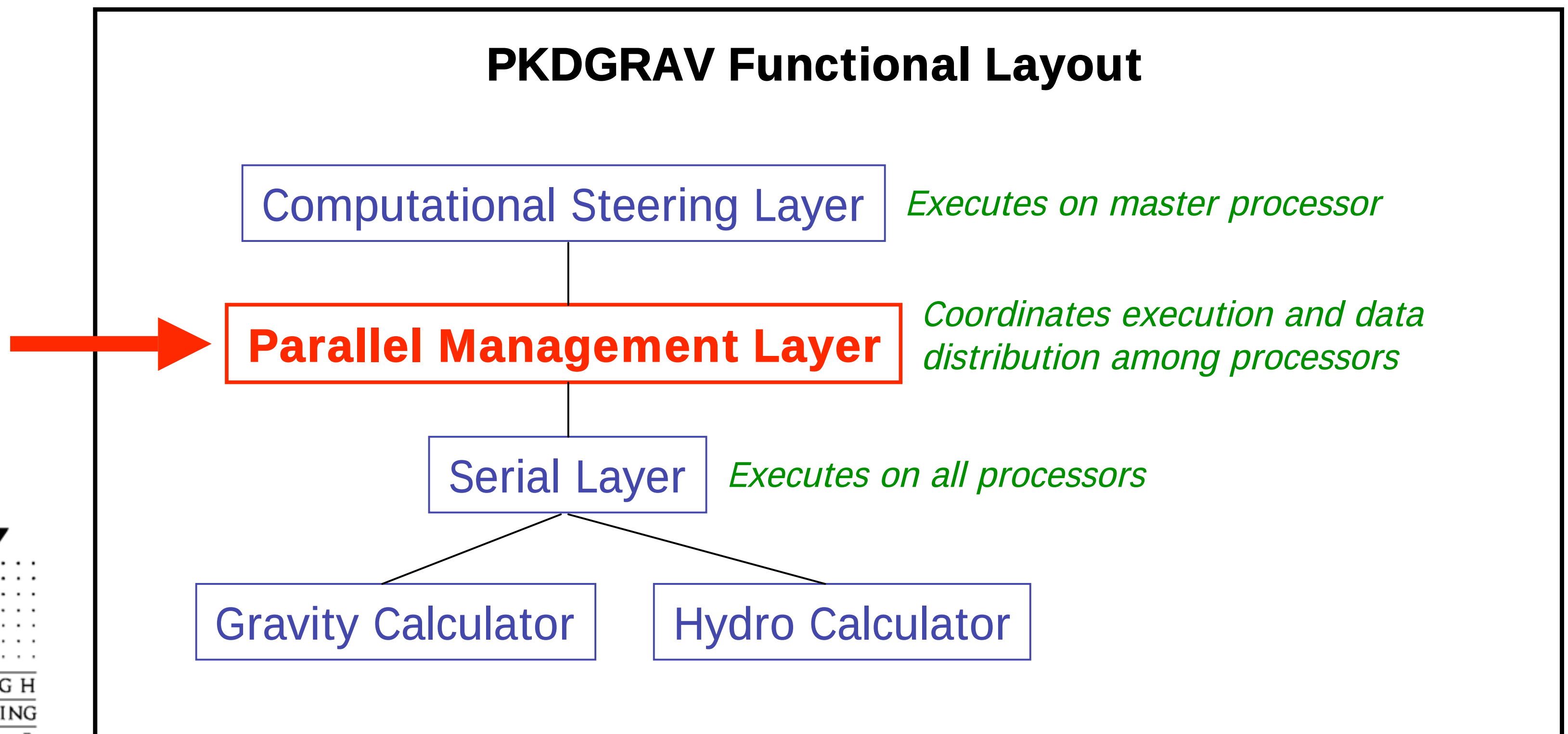


PITTSBURGH
SUPERCOMPUTING
CENTER

Proof of Concept: PHASE 2
N Tropy

(Currently in progress)

- Use only Parallel Management Layer of PKDGRAV.
- Rewrite everything else from scratch



PITTSBURGH SUPERCOMPUTING CENTER

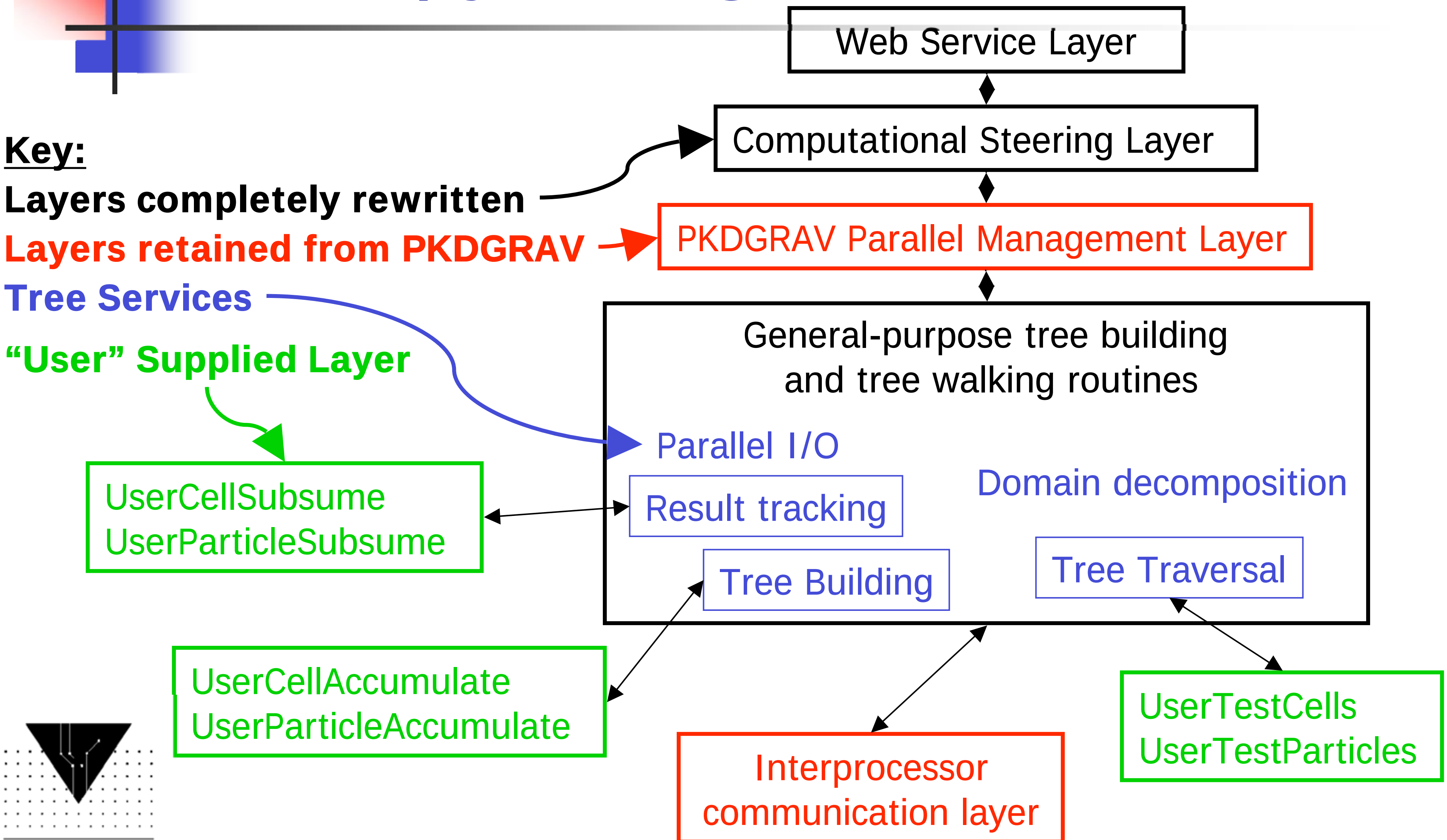
Proof of Concept: PHASE 2
N Tropy
(Currently in progress)

- Use only Parallel Managment Layer of PKDGRAV.
- Rewrite everything else from scratch
- PKDGRAV benefits to keep:
  - Flexible client-server scheduling architecture
    - Threads respond to service requests issued by master.
    - To do a new task, simply add a new service.
  - Portability
    - Interprocessor communication occurs by high-level requests to “Machine-Dependent Layer” (MDL) which is rewritten to take advantage of each parallel architecture.
  - Advanced interprocessor data caching
    - < 1 in 100,000 off-PE requests actually result in communication.

PITTSBURGH SUPERCOMPUTING CENTER

N Tropy Design

2-Point and 3-Point algorithm are now complete!



N Tropy "Meaningful" Benchmarks

- The purpose of this framework is to minimize development time!
- Rewriting user and scheduling layer to do an N-body gravity calculation:

N Tropy "Meaningful" Benchmarks

- The purpose of this framework is to minimize development time!
- Rewriting user and scheduling layer to do an N-body gravity calculation:

3 Hours

PITTSBURGH SUPERCOMPUTING CENTER

N Tropy New Features
(coming soon)

- Dynamic load balancing
  - Workload and processor domain boundaries can be dynamically reallocated as computation progresses.
- Data pre-fetching
  - Predict request off-PE data that will be needed for upcoming tree nodes.
  - Work with CMU Auton-lab to investigate active learning algorithms to prefetch off-PE data.

PITTSBURGH
SUPERCOMPUTING
CENTER

N Tropy New Features
(coming soon)

- Computing across grid nodes
  - Much more difficult than between nodes on a tightly-coupled parallel machine:
    - Network latencies between grid resources 1000 times higher than nodes on a single parallel machine.
  - Nodes on a far grid resources must be treated differently than the processor next door:
    - Data mirroring or aggressive prefetching.
    - Sophisticated workload management, synchronization

PITTSBURGH
SUPERCOMPUTING
CENTER

Conclusions

- Most data analysis in astronomy is done using trees as the fundamental data structure.
- Most operations on these tree structures are functionally identical.
- Based on our studies so far, it appears feasible to construct a general purpose parallel framework that users can rapidly customize to their needs.

PITTSBURGH SUPERCOMPUTING CENTER