Simulations, Data Analysis & High Performance Computing After Planck

Julian Borrill, Reijo Keskitalo & Ted Kisner

Computational Cosmology Center, Berkeley Lab

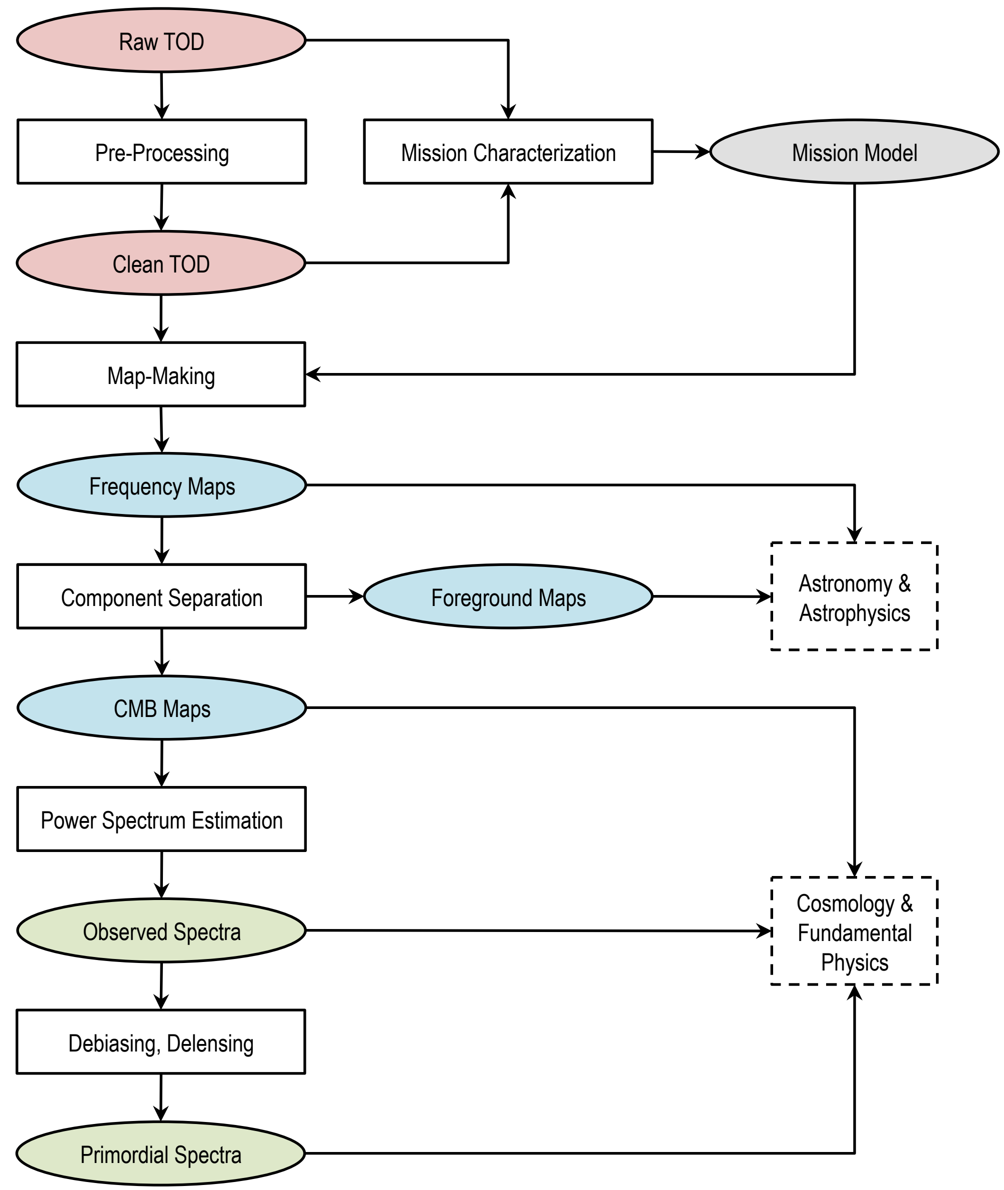
& Space Sciences Laboratory, UC Berkeley

The scientific results that we present today are a product of the Planck Collaboration, including individuals from more than 100 scientific institutes in Europe, the USA and Canada.

Planck is a project of the European Space Agency, with instruments provided by two scientific Consortia funded by ESA member states (in particular the lead countries: France and Italy) with contributions from NASA (USA), and telescope reflectors provided in a collaboration between ESA and a scientific Consortium led and funded by Denmark.

Data Analysis

- DA challenge comes from both systematic and statistical uncertainties.
- DA pipeline is an alternating sequence of
    a) domain-specific systematic mitigation
    b) S/N-increasing data compression
- Must propagate both data *and their covariance* for a sufficient statistic.



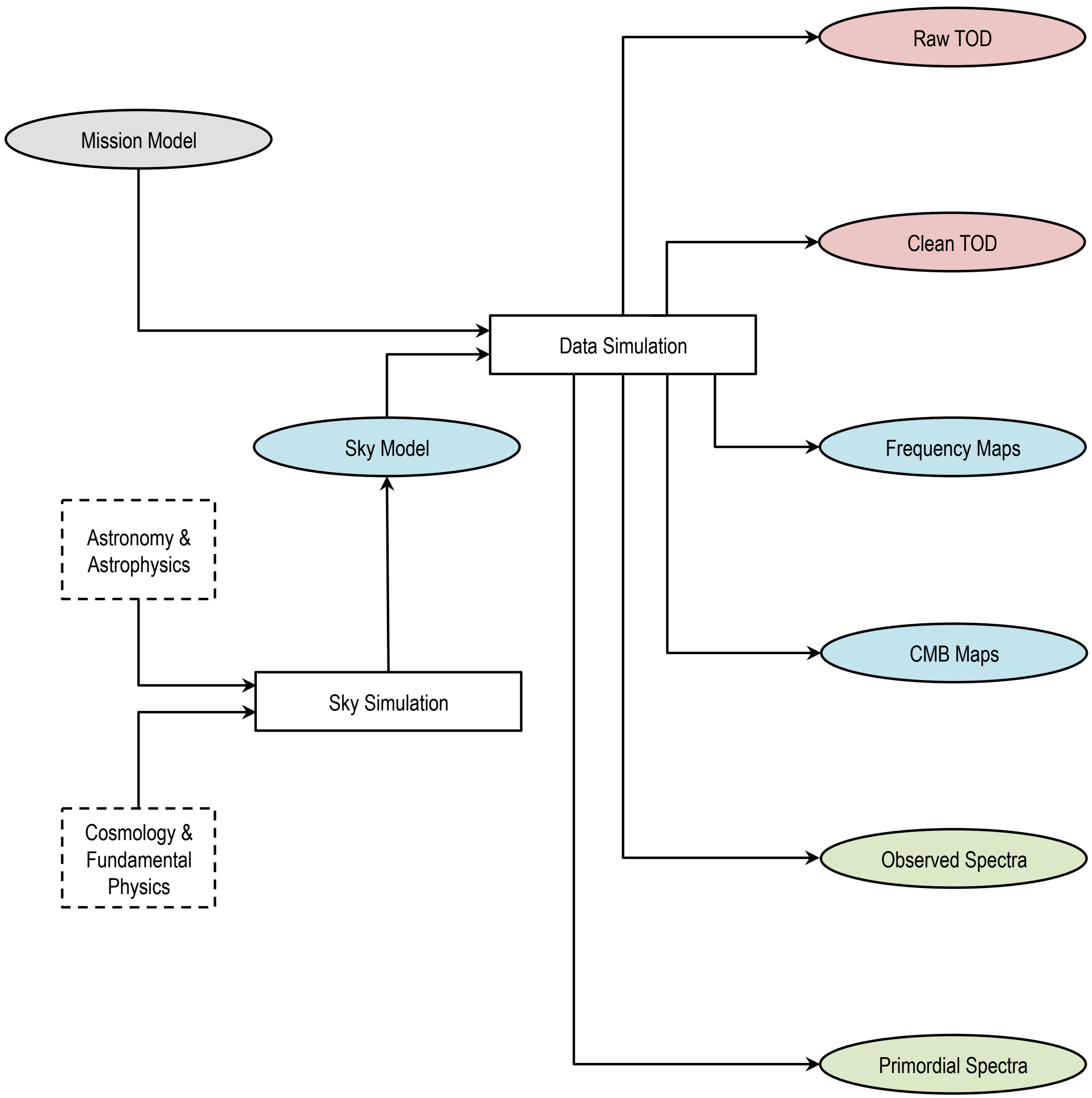
Analysis Methods

- CMB data volumes:
  - Time domain: $\mathcal{N}_t \sim \Sigma_{det}$ Sampling Rate (Hz) x Observation Time (s)
  - Pixel domain: $\mathcal{N}_p \sim \Sigma_{freq,\ pol}\ 10^9$ x Sky Fraction / [Beam (arcmin)]^2
- CMB data analysis scaling dominated by:
  - $\mathcal{N}_p^{\,3}$ for exact methods with explicit covariance matrices.
  - $\mathcal{N}_{mc}\,\mathcal{N}_t$ for approximate methods with MC uncertainty quantification.
- Computational constraints (1% cycles/year on Top 10 system):
  - 2000 : $\mathcal{N}_p < 10^6$ & $\mathcal{N}_t < 10^{12}$
  - 2015 : $\mathcal{N}_p < 10^7$ & $\mathcal{N}_t < 10^{15}$
  - 2030 : $\mathcal{N}_p < 10^8$ & $\mathcal{N}_t < 10^{18}$

Assumes:
– Moore's Law
– 100% & 1% efficiency

- Except in special cases, exact methods now computationally intractable.

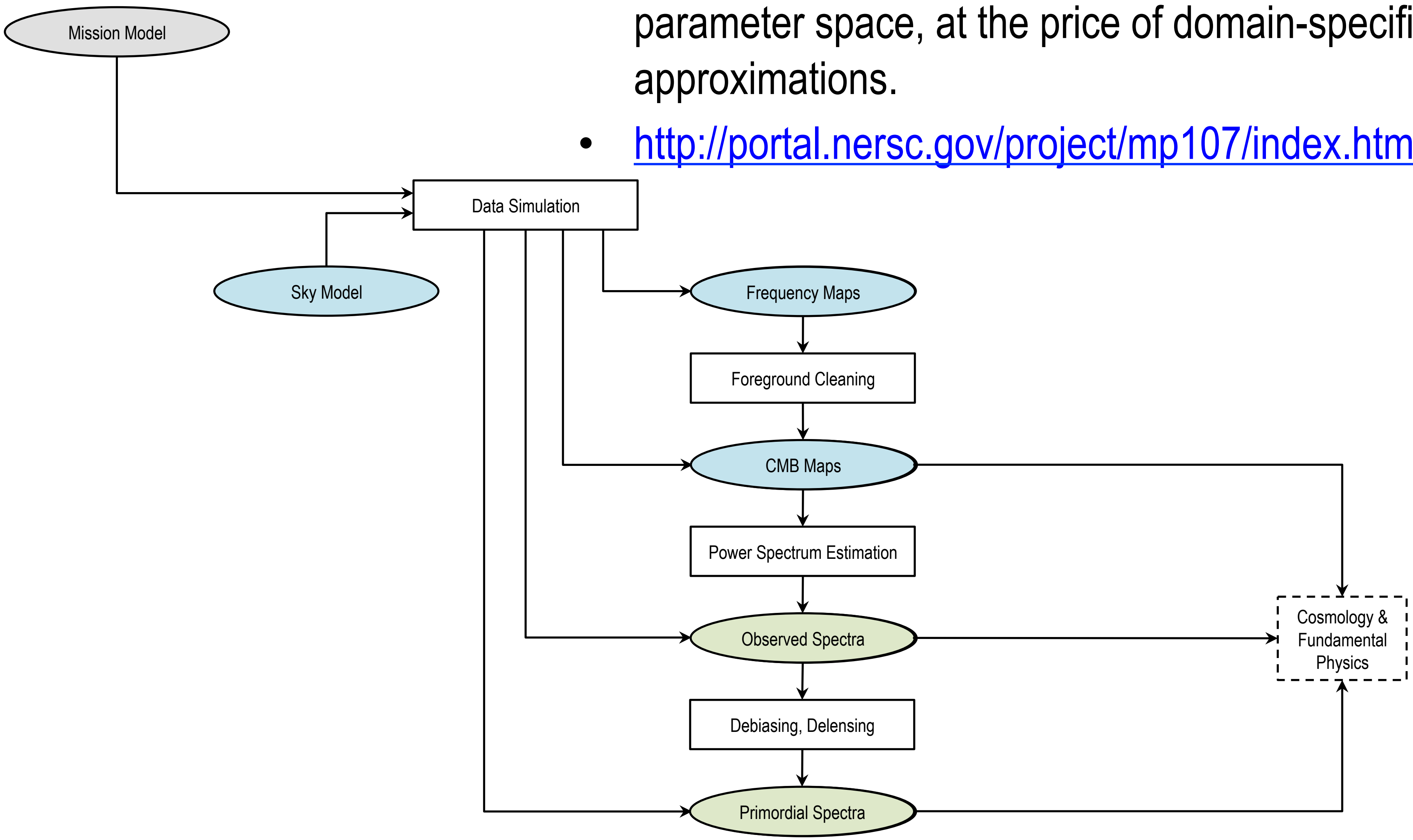
Simulations



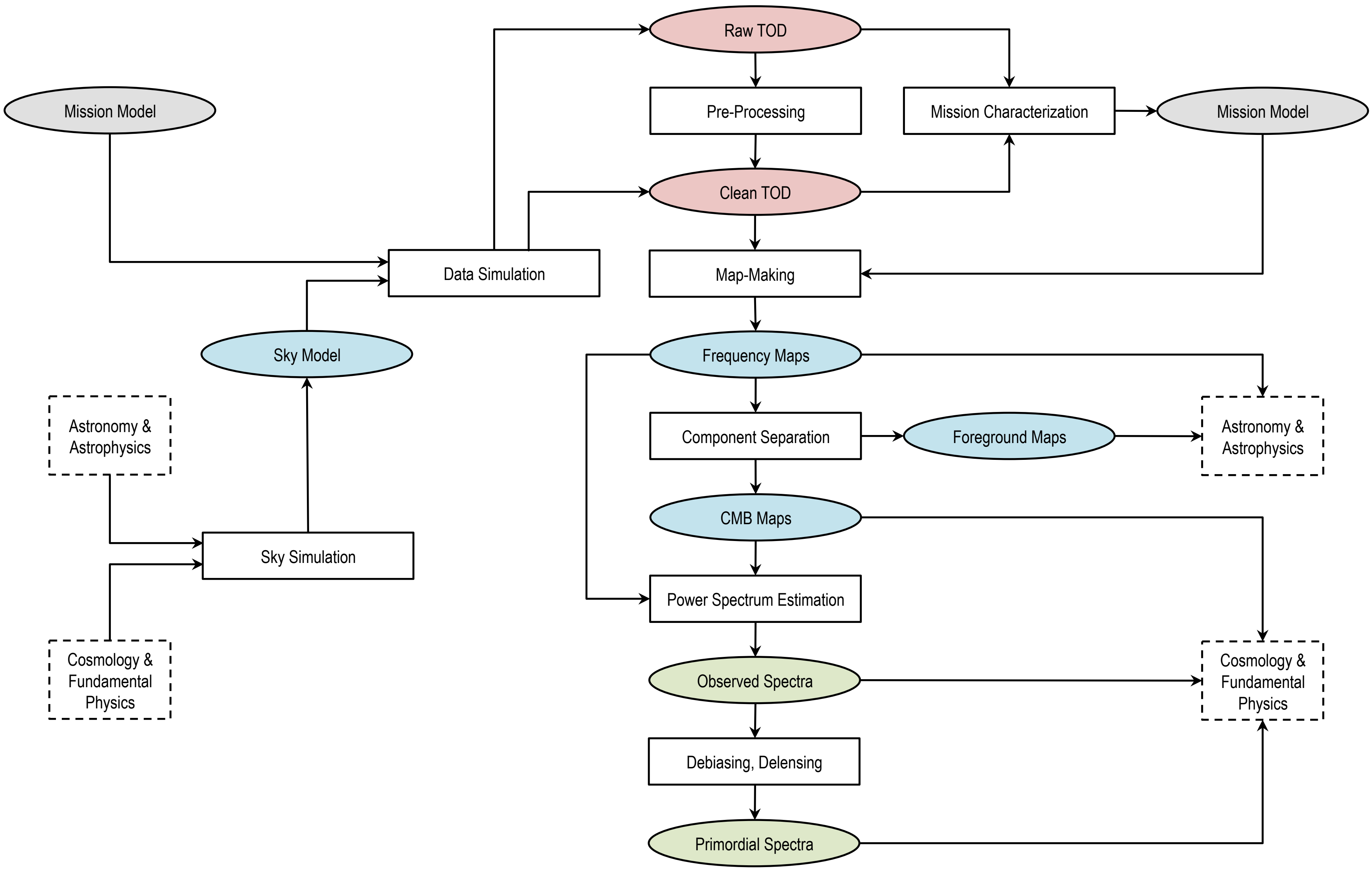
- Needed for:
  - Forecasting
  - Mission design & development
  - DA validation & verification
  - Data uncertainty quantification & debiasing (MC)
- From top to bottom, trade-off between:
  - computational cost
  - realism/reliability

Forecasting SimDA

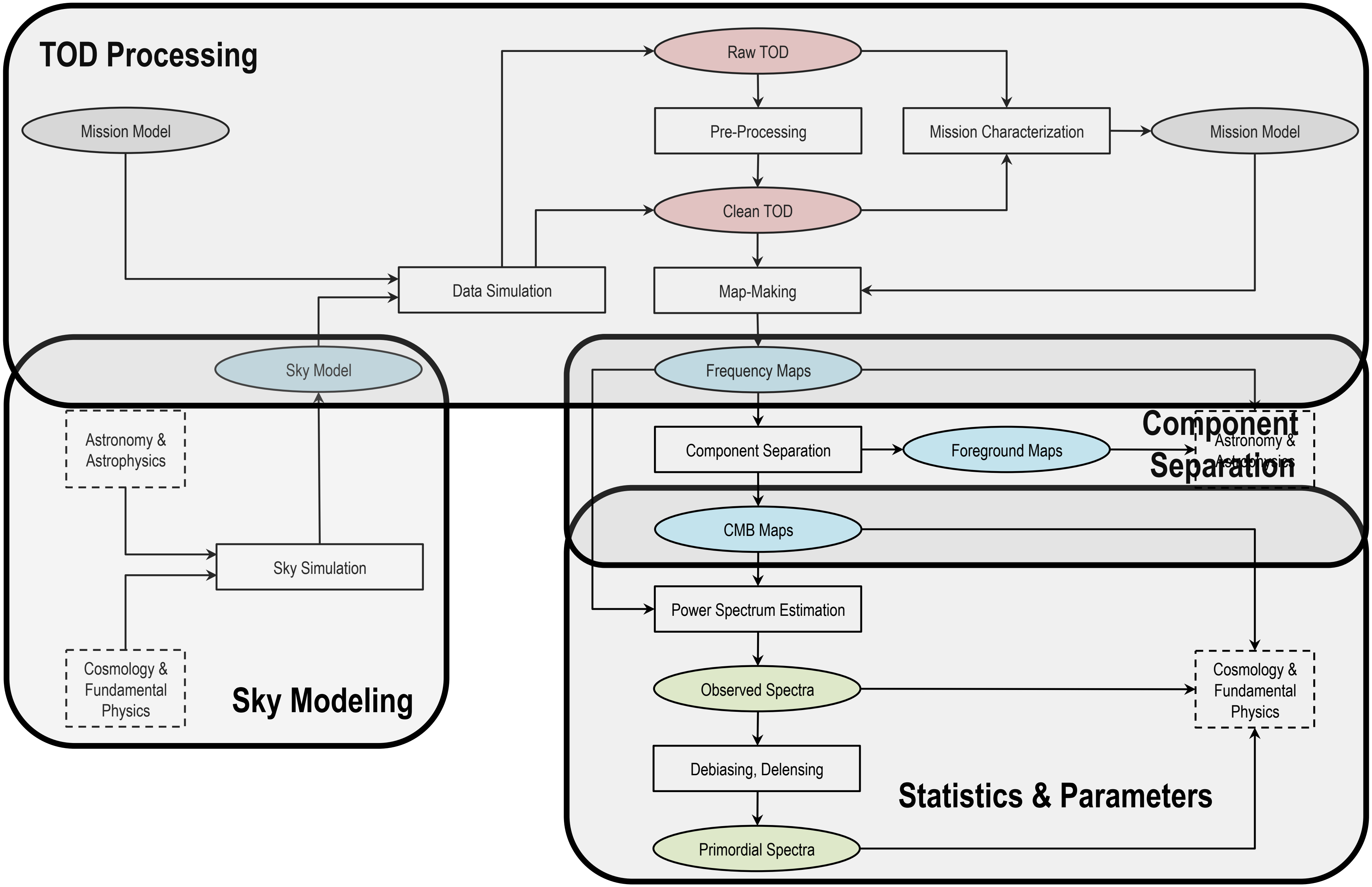
- Speed allows for exploration of full mission parameter space, at the price of domain-specific approximations.
- http://portal.nersc.gov/project/mp107/index.html



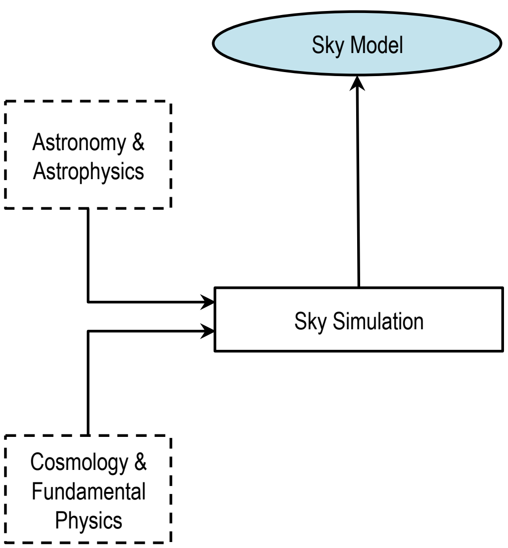
Production SimDA With Feedback



SimDA: Sub-Domains



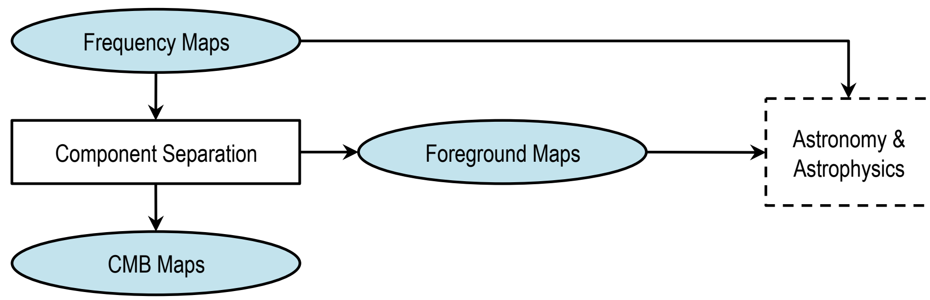
Sky Modeling



- Key Challenges:
  - Reliability: noisy, confused, band-passed, beam-convolved templates (inc. Planck) and/or speculative modeling.
  - Self-consistency: eg. CMB secondaries & extra-Galactic foregrounds
  - Usability: software engineering

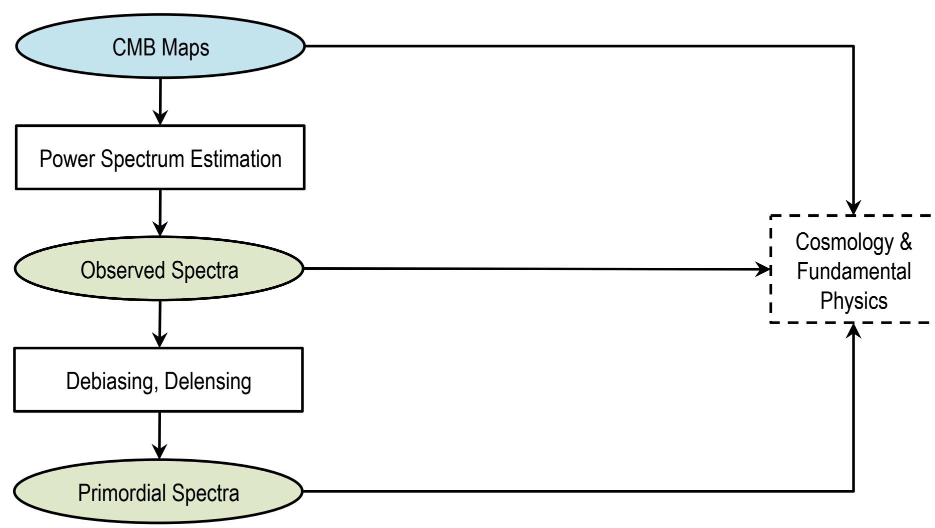
Component Separation

- Key Challenges:
  - Validation: are these the right algorithms for the (as yet unknown) real foregrounds?
  - Verification: are these algorithms right given (as yet flawed) simulated foregrounds?
  - Polarization!

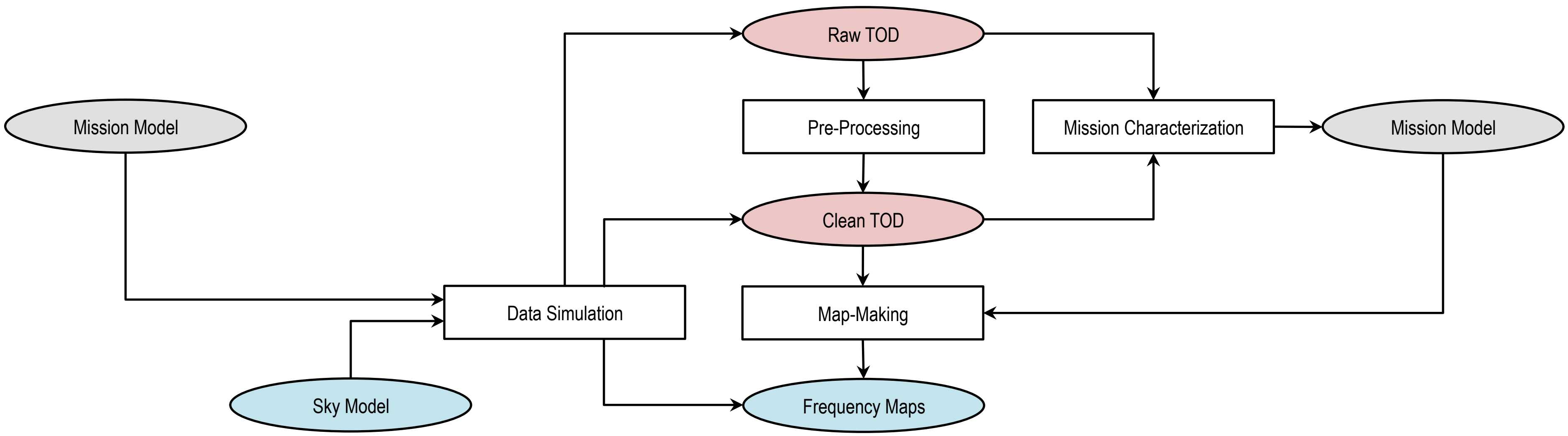


Statistics & Parameters

- Key Challenges:
  - Reliability: sufficiency of data covariance approximations.
  - Tractability: disk space for many millions of MC maps.

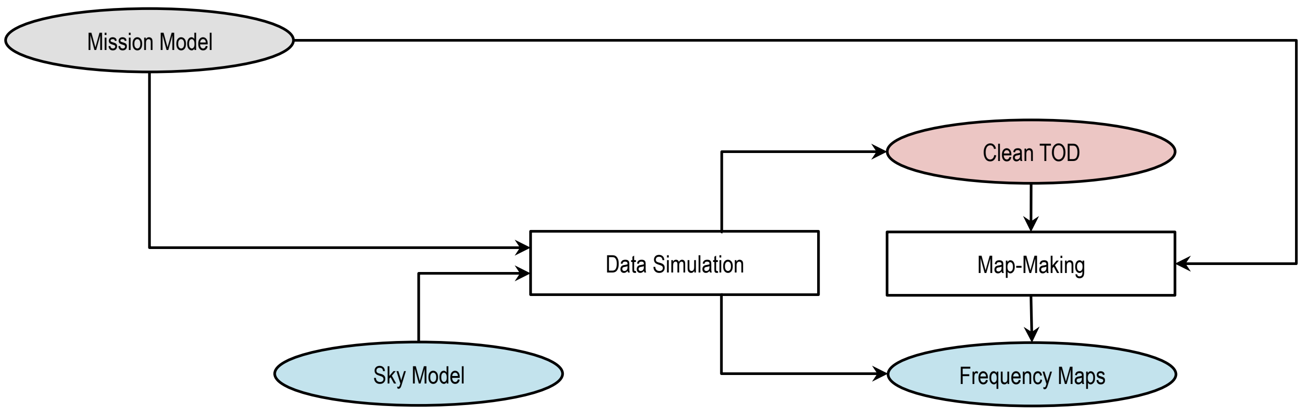


TOD Challenges



- Two bounding challenges:
  - *Tractability* for massive Monte Carlo sets.
  - *Usability* for exploratory pre-processing & mission characterization.

Massive Monte Carlos



- Operation count scales with
  - Number of MC realizations: $\mathcal{N}_{mc} \sim 10^4$
  - Number of map-makings per realization: $\mathcal{N}_{mm} \sim 10$
  - Number of PCG iterations per map-making: $\mathcal{N}_{it} \sim 10$
  - Number of operations per PCG iteration: $\mathcal{N}_{ops} \sim 10 \times \mathcal{N}_t$
- Required FLOP $\sim 10^7\ \mathcal{N}_t \sim 10^{19}$ for Planck

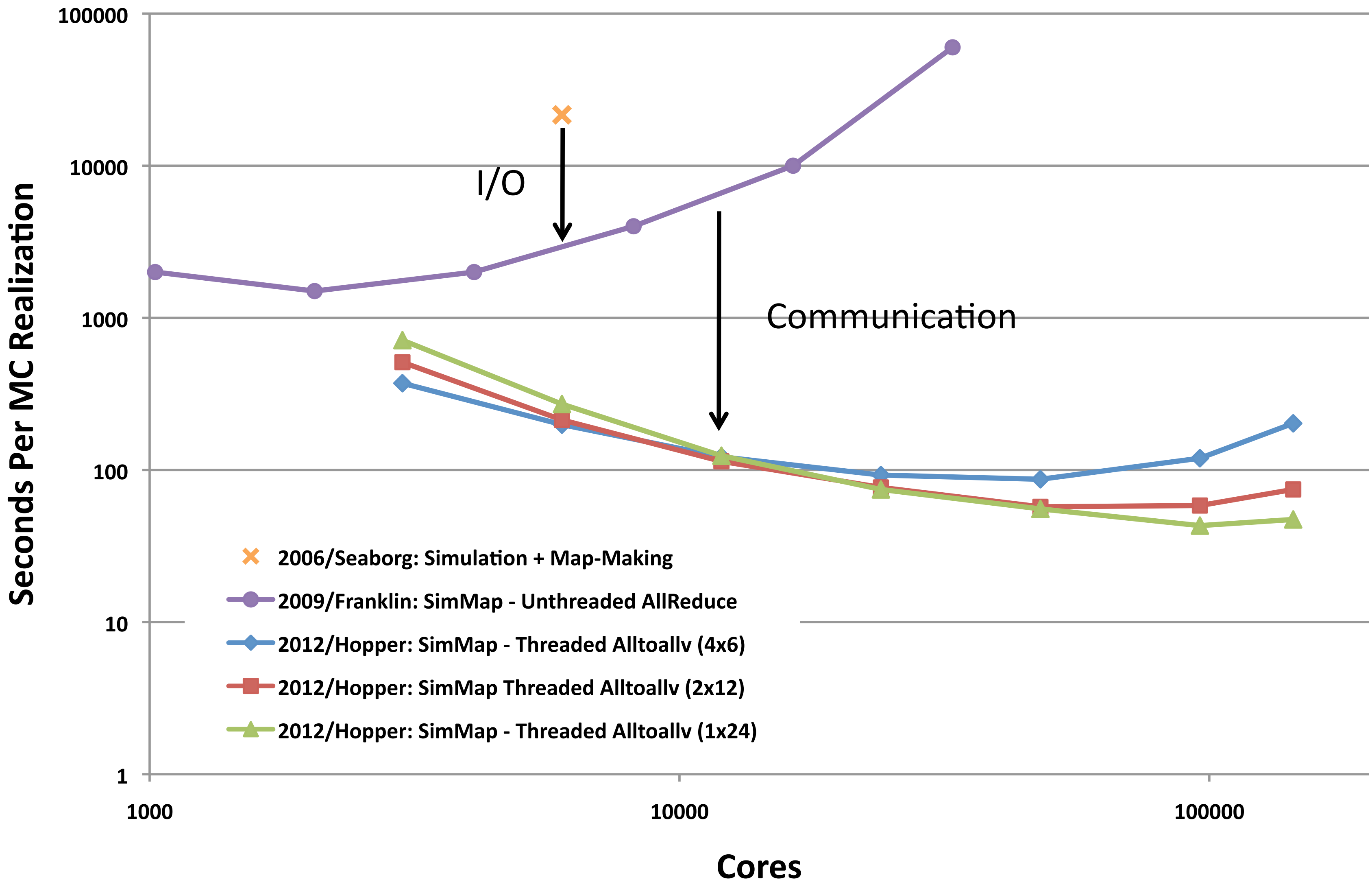
High Performance Computing

- 10^{19} FLOP ~ 10^5 CPU-years at 1% efficiency on 1GHz CPU

  ⇒ Massive parallelism + Moore's Law growth
- Whole-data reduction

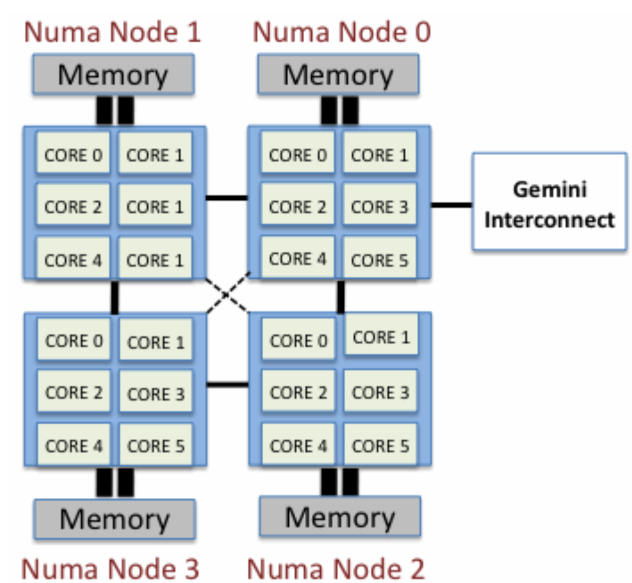
  ⇒ Tightly-coupled cores (not grid/cloud/at-home/etc)
- Planck solution:
  - NERSC: Open-access HPC facility with long-term system upgrade plan.
    - New Top 10 system every 2-3 years
    - 6,000 users from 50 countries
  - NASA/DOE MOU guaranteed minimum annual NERSC allocation for mission lifetime:
    - In practice 1% NERSC cycles/year ~ 10^5 x Peak FLOP/s

Implementation/Architecture Evolution

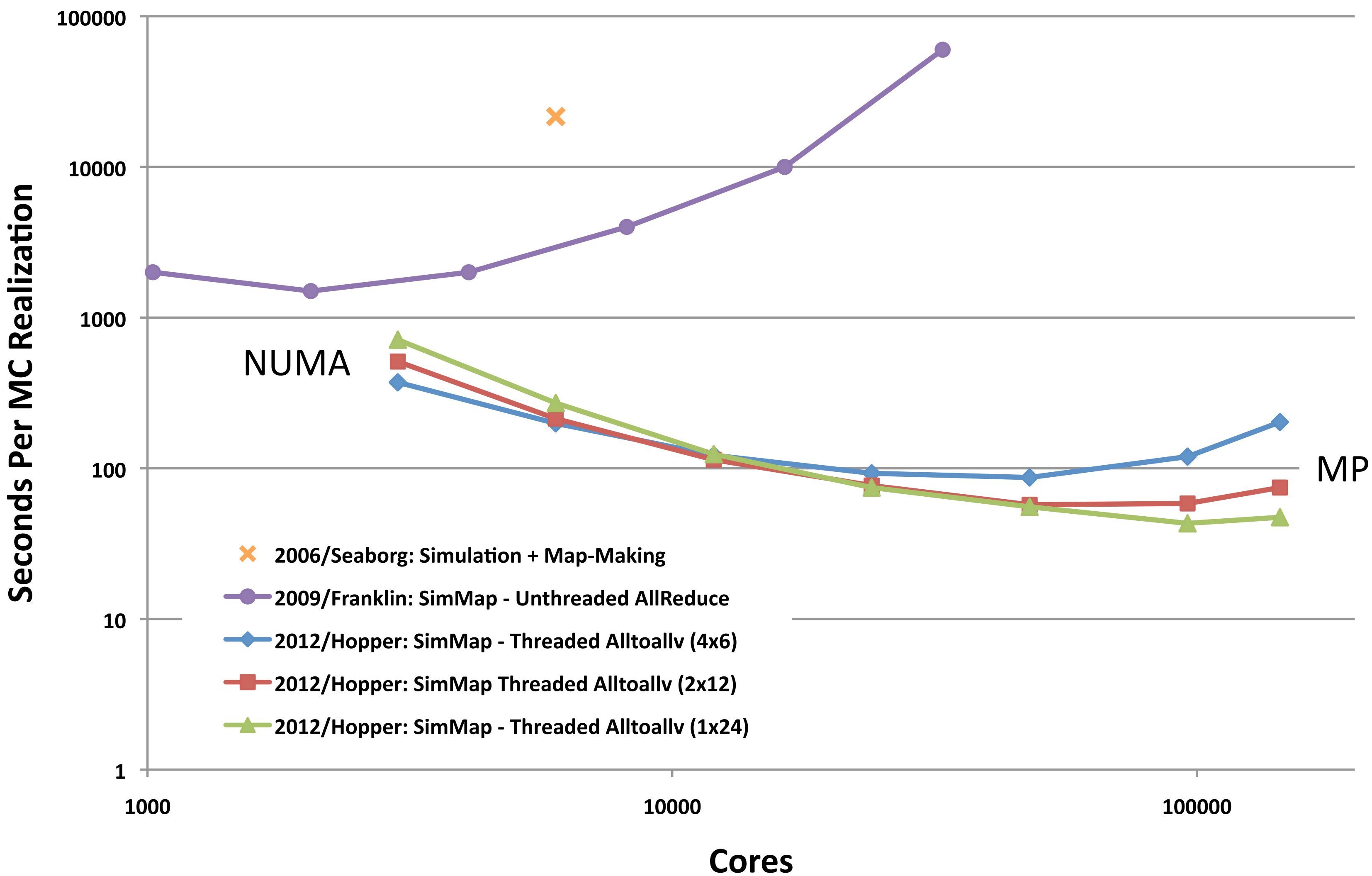


Architecture Evolution

- Clock speed is no longer able to maintain Moore's Law.
- Many-core and GPU are two major approaches.
- Both of these will require
  - significant code development
  - performance experiments & auto-tuning
- Eg. NERSC's Cray XE6 system *Hopper*
  - 6384 nodes
  - 2 sockets per node
  - 2 NUMA nodes per socket
  - 6 cores per NUMA node
- What is the best way to run hybrid code on such a system?



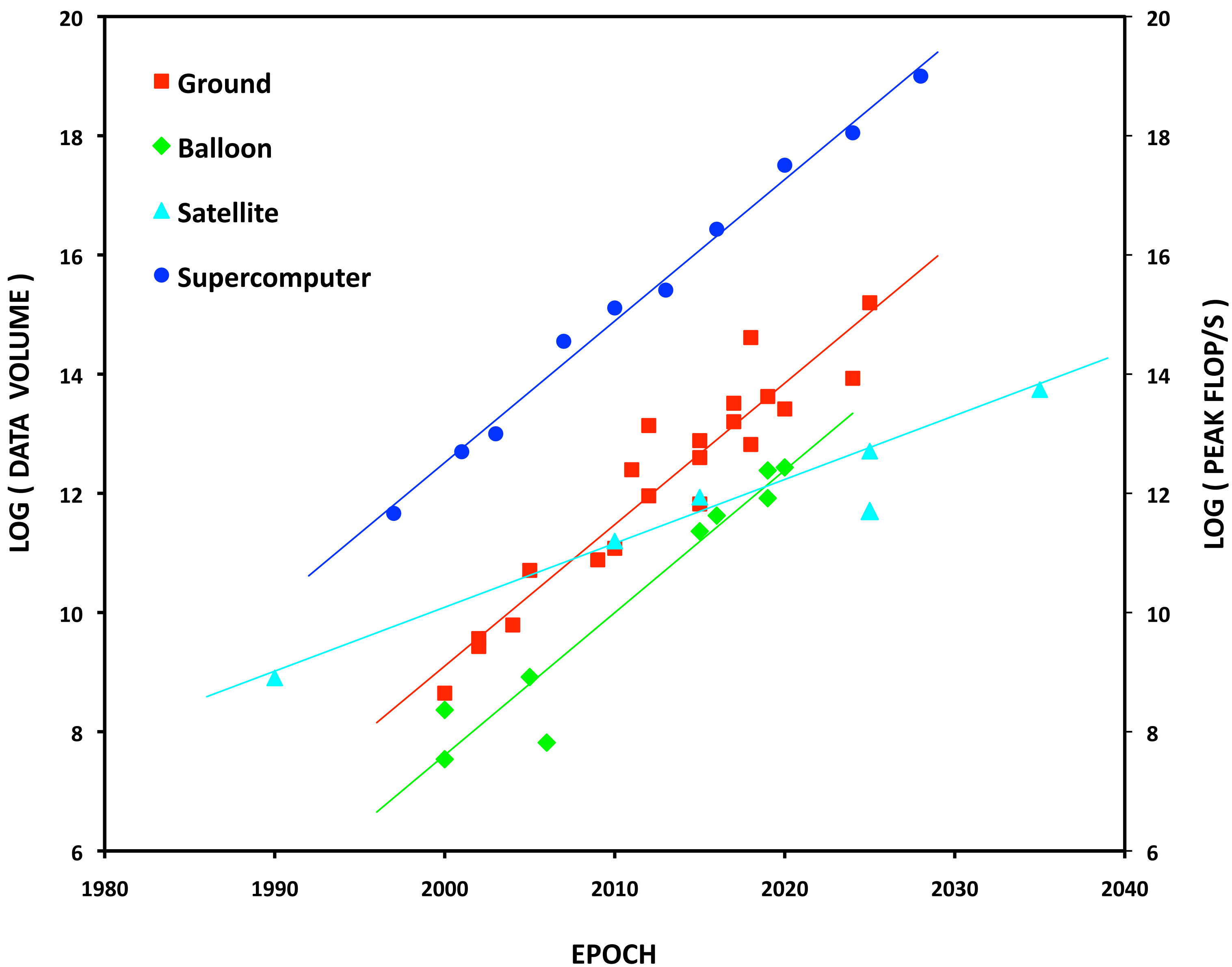
Configuration With Concurrency



Results: Planck Full Focal Plane 8

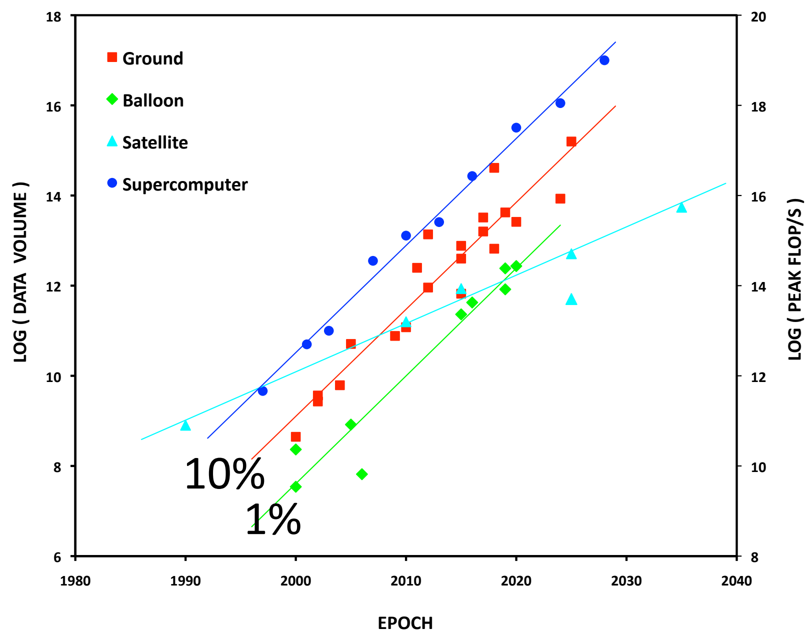
- 10^4 Monte Carlo realizations reduced to 10^6 maps
  - multiple maps made per simulation

Data & HPC Growth



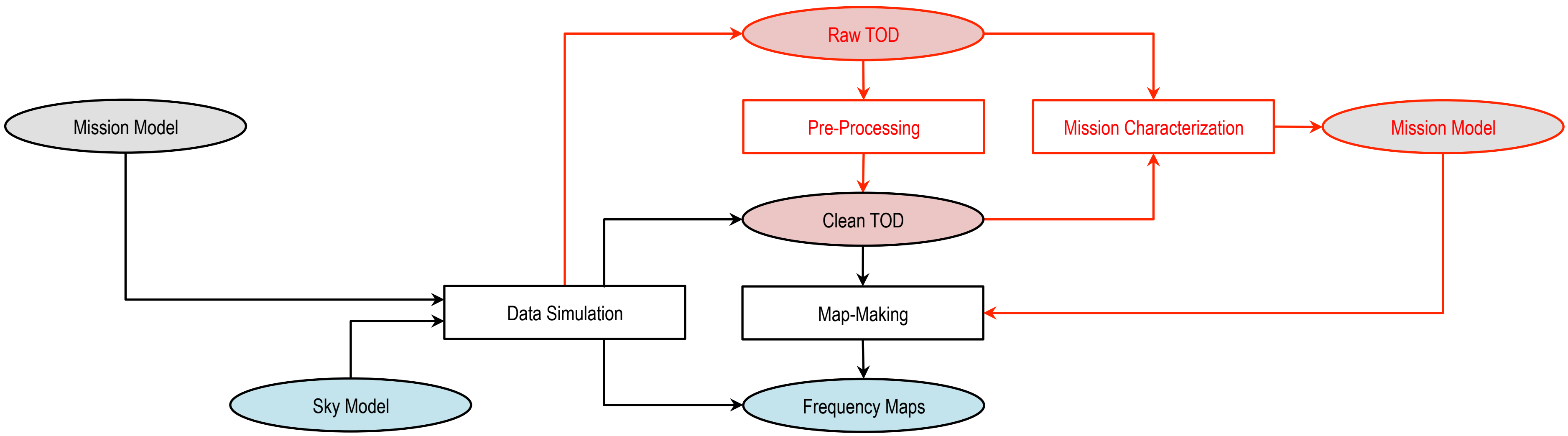
Next Generation Challenges

- Computational Efficiency
  - Required FLOP ~ $10^7 \mathcal{N}_t$
  - Available FLOP ~ 10^5 x Peak
  - Efficiency: $\varepsilon > 10^2 \mathcal{N}_t$ / Peak
    - compare suborbital & space!



- Next-generation HPC challenges
  - Energy constraints limiting Watt/FLOP (Tianhe-2 ~ Belize!)
  - More complex architectures will be harder to program efficiently
    - system heterogeneity, deep memory hierarchies, dark silicon, etc
  - End of Moore's Law

Pre-Processing & Mission Characterization



- A limiting factor for Planck has been our ability to easily and quickly
  - simulate detector-level data in full detail
  - prototype pre-processing/mission characterization algorithms.
- As sensitivity increases, mitigating systematics and characterizing their residuals becomes ever more important.

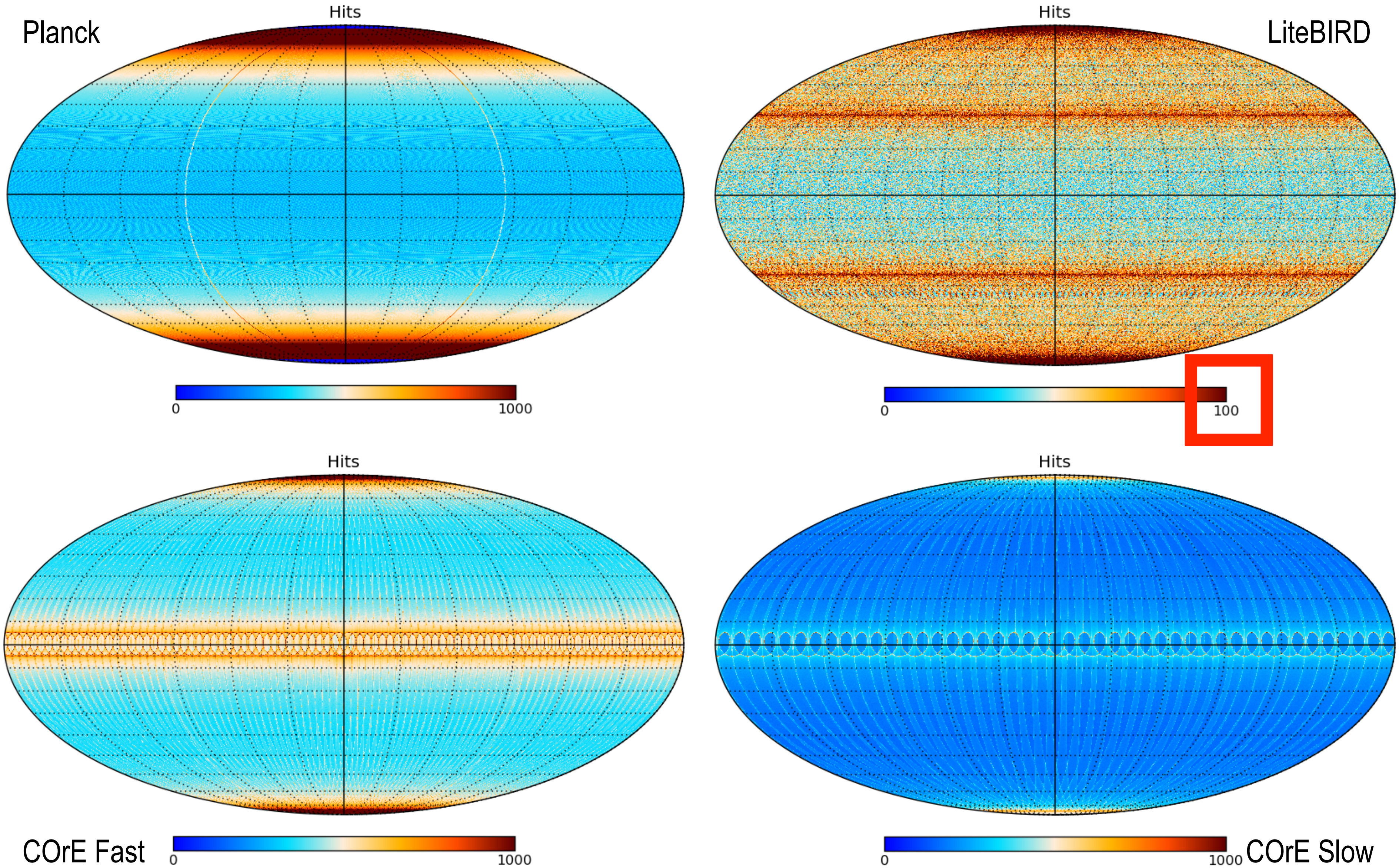
TOAST Overview

- Competing requirements:
  - Massively parallel & very efficient even on coming HPC architectures
  - Easy for non-HPC experts to adapt, extend & run
- Re-implement entire TOAST framework as open source python modules
  - Expanded developer base
  - Rapid prototyping
  - Split generic and experiment-specific elements
- Efficiency issues:
  - Start-up cost: pre-bundle libraries (eg. pyinstaller)
  - I/O avoidance: pass data between modules in memory
  - Compute efficiency: link to compiled C(++) code at key points

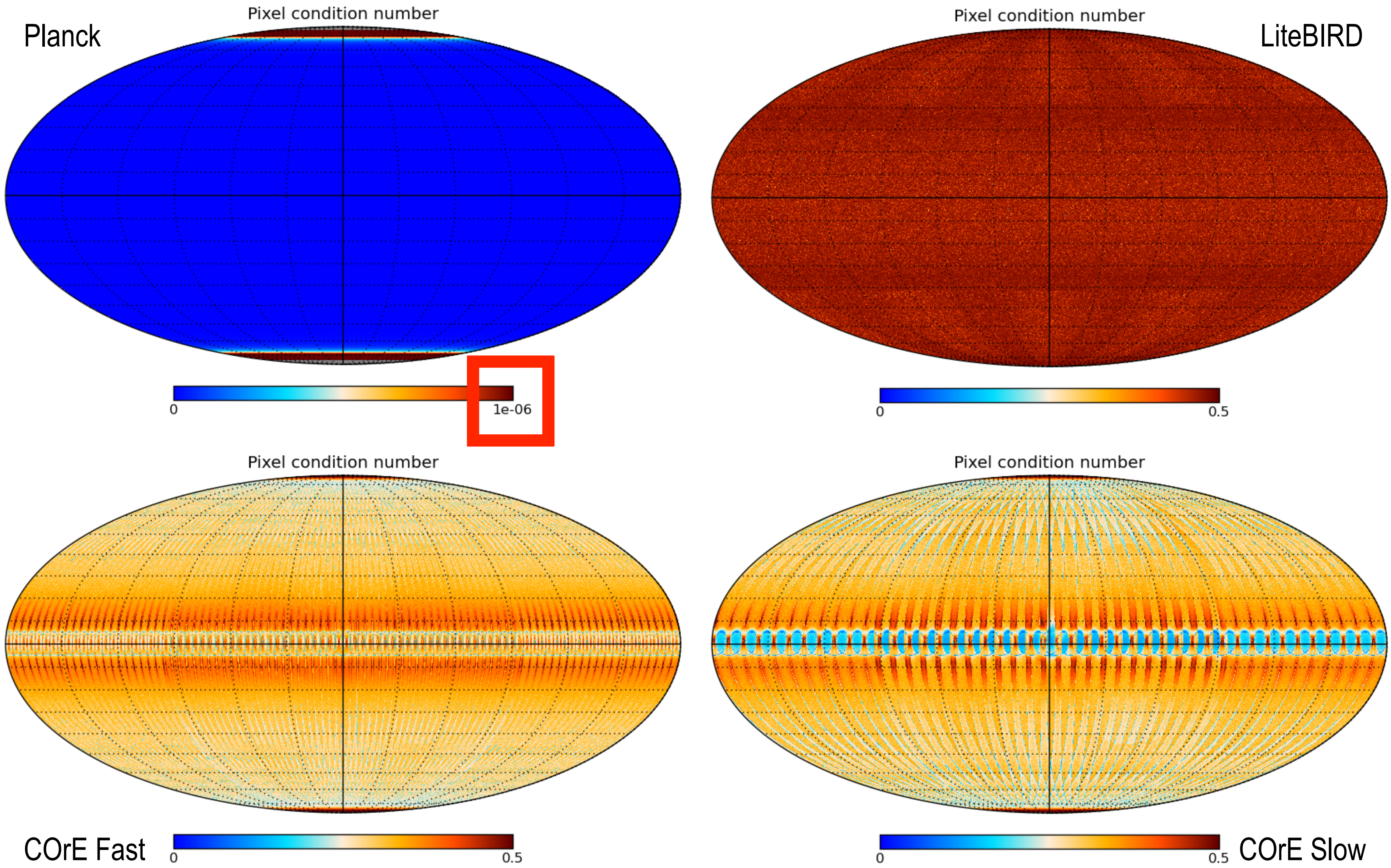
Example: Single-Detector Maps

- Single-detector maps provide powerful systematics tests since they avoid beam, bandpass mismatch issues.
  - Also provide checks on single-detector systematics (side-lobes etc)
- Polarized single-detector maps require observations of each pixel with many attack angles.
- Comparing scanning strategies is an inherently time-domain activity.
- Satellite scans parameterized by precession & spin angles & rates.
- Compare 4 cases:
  - Planck
  - LiteBIRD (with HWP)
  - COrE fast spin
  - COrE slow spin

Example: Single-Detector Hit Maps

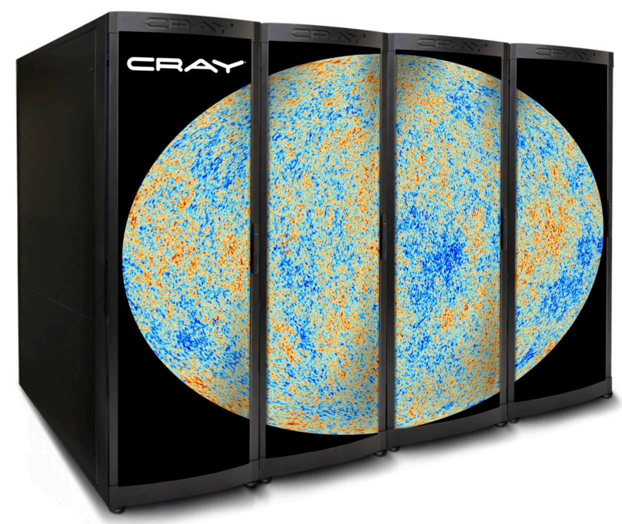


Example: Single-Detector Condition Maps

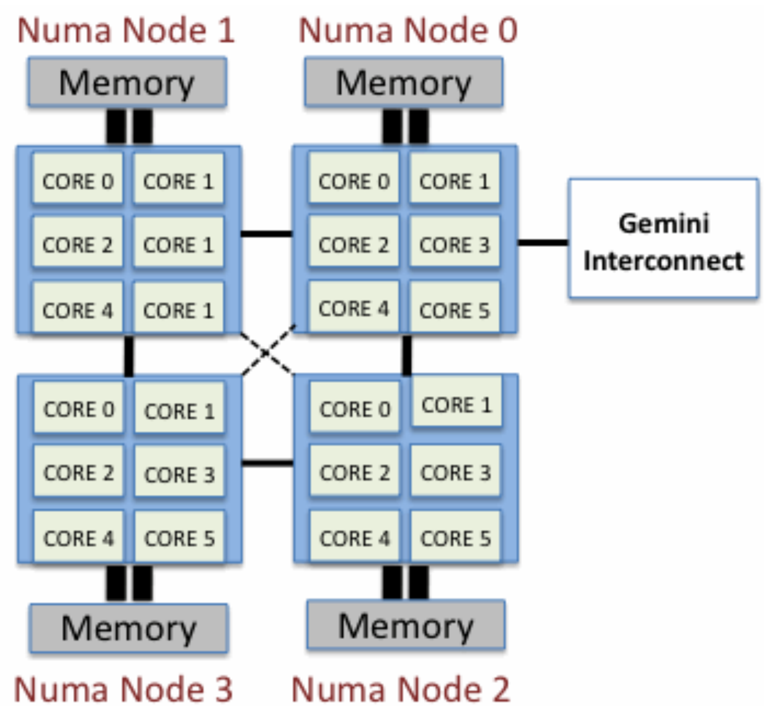


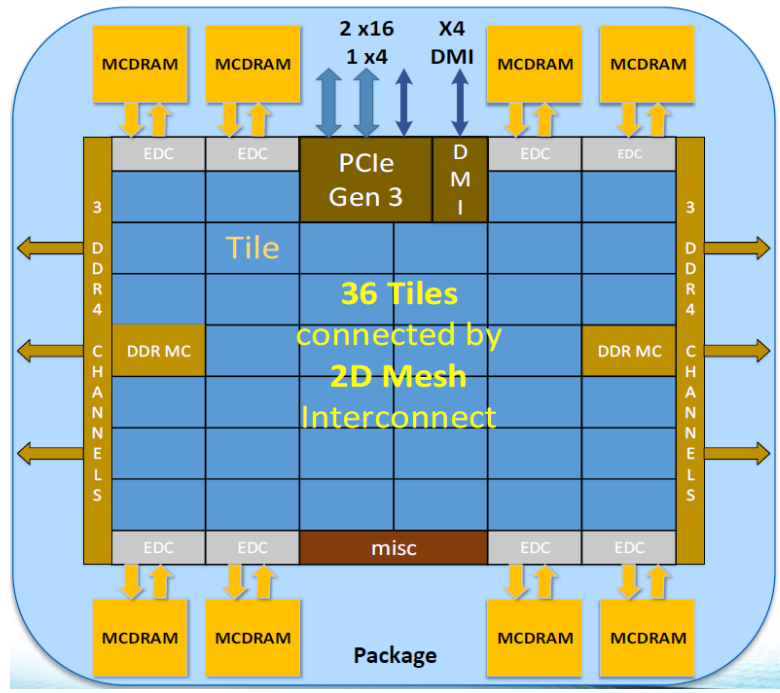
TOAST Status

- Base framework & generic tools/scripts
  - Public git repo https://github.com/hpc4cmb/toast
- Experiment-specific extensions & scripts:
  - Private git repos https://github.com/hpc4cmb/toast-X
  - X: toast-planck, toast-litebird, toast-core, toast-cmbs4, etc
- Planned additions/extensions:
  - **Xeon Phi KNL port/optimization**
  - On-the-fly band-pass integration
  - HWP-varying beam, bandpass
  - Multichroic/multiplexed cross-talk
  - Planet/variable source observations
  - Atmosphere & ground-pickup



Energy-Constrained Node Evolution





Magny-Cours (24 threads) => Knights Landing (~160 threads)

A Modest Proposal

- A two-tier community-wide program:
  - developing common, generic capabilities in the public domain
  - deploying them for specific analyses within our various collaborations