

気候・気象シミュレーションとHPC： 京における全球雲解像モデルNICAMの取り組みを通して

八代 尚

(独)理化学研究所 計算科学研究機構

Hisashi YASHIRO

RIKEN Advanced Institute of Computational Science



自己紹介

- 八代 尚 (Hisashi Yashiro)
 - 出身：青森県弘前市
 - 出身大学：東北大学大学院 理学研究科 地球物理学専攻
 - 研究テーマ：温室効果気体と温室効果に関わる気体の観測的研究
 - 博士論文：対流圏一酸化炭素の時空間変動に関する研究（理学博士）

Measurement system for trace gases



Ny-Ålesund, Svalbard, Norway (79°N, 12°E)



自己紹介

- 八代 尚 (Hisashi Yashiro)

- 職歴

- 2008/11～2011/9 (独)海洋研究開発機構 地球環境フロンティア研究センター
大気組成変動予測研究プログラム ポストドクトラル研究員
- 2011/10～2012/3 (独)理化学研究所 計算科学研究機構
複合系気候科学研究チーム 特別研究員
- 2012/4～現在 同 研究員

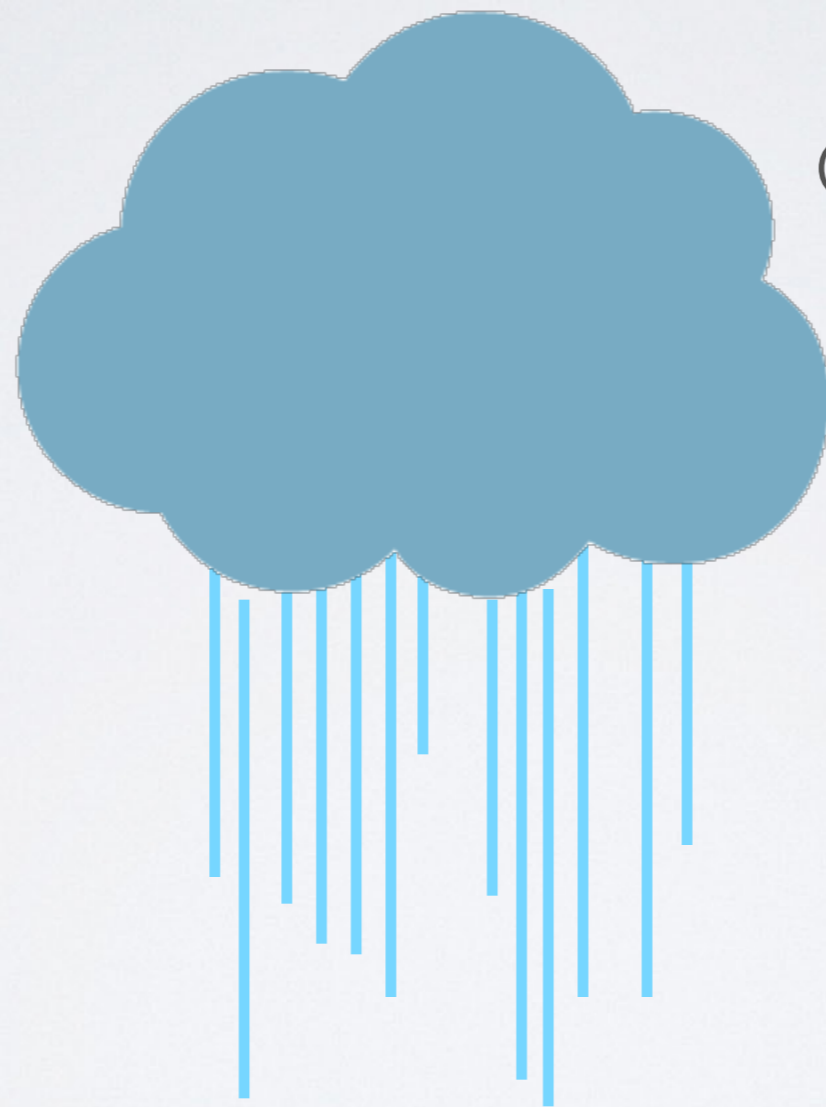
- 研究テーマ

- 化学気候モデルを用いた大気微量気体の時空間変動・全球収支の研究
- 高解像度数値気象モデルの開発
 - 気象シミュレーションのブレークスルーとなり得る大規模計算への挑戦
 - 最新の計算機トレンドを取り入れた気象計算技術基盤の確立

本日の内容



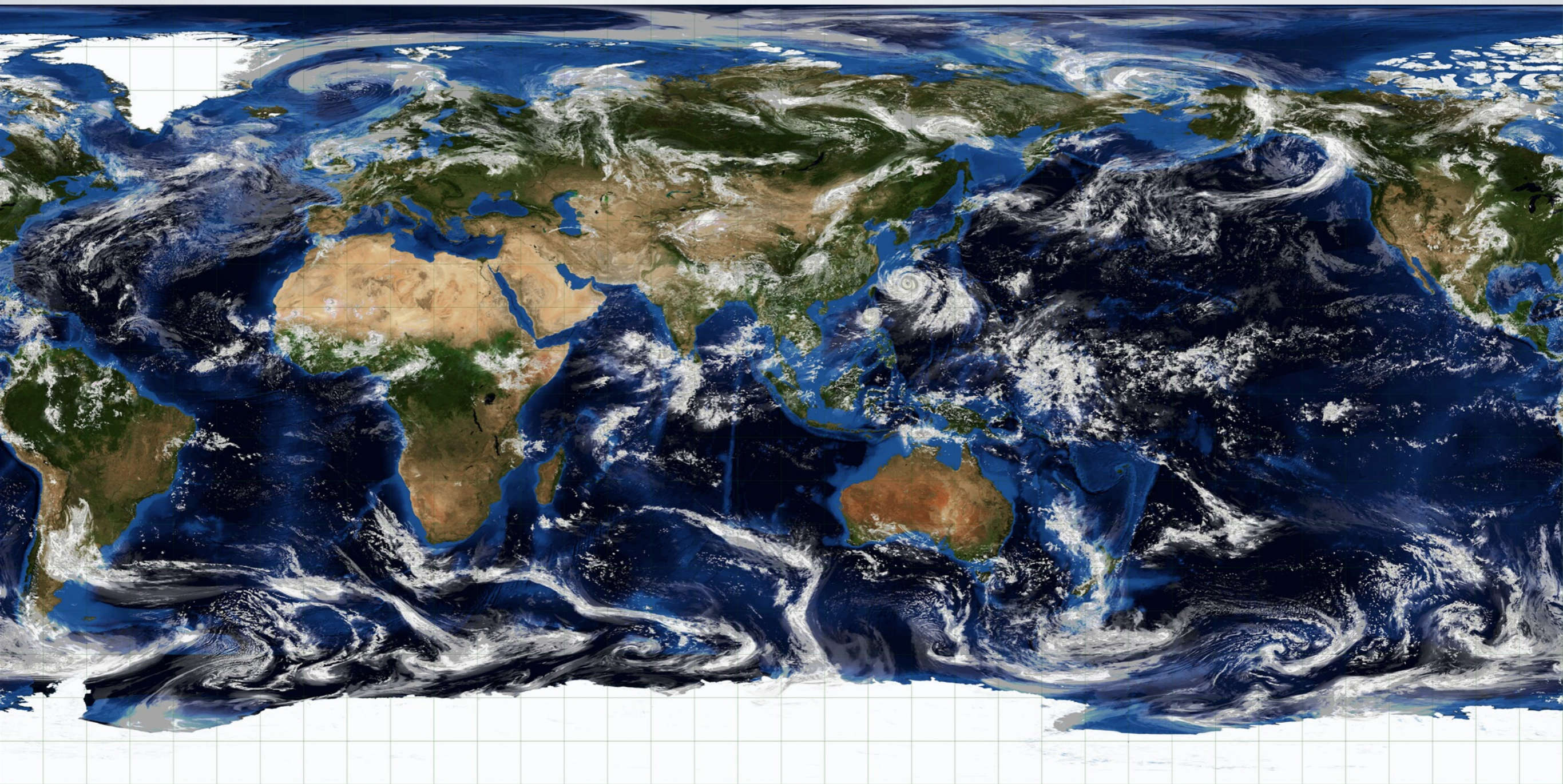
本日の内容



Computing of the cloud

全球の雲の姿

Visualized by 吉田龍二 (RIKEN AICS)



1 kmを切る解像度の全球大気シミュレーション

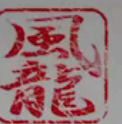
20480nodes(163840cores) on the K computer

Visualized by 吉田龍二 (RIKEN AICS)

NICAM 870 m - 96 levels
Real Case Simulation: 25 - 26, Aug., 2012

SPIRE field-3: Study of extended-range predictability using GCSRAM

RIKEN / AICS: Computational Climate Science Research Team



講義の流れ

- * 気象学・気候学への招待
- * 気象モデルシミュレーションの基礎
- * 京における全球雲解像モデルNICAMの性能評価
- * 今後に向けた取り組み

気象学・気候学への招待



身近にある気象現象

日本経済新聞 8月24日 土曜日 English 中文

Web刊 速報 ビジネスリーダー マーケット マネー テクノロジー ライフ スポーツ 朝

トップ 特集 コラム 読者アンケート 紙面連動 社説・春秋 映像 アジアBiz ウーマン

高知・四万十市で41.0度 国内観測史上最高を記録

2013/8/12 14:09

小 中 大 保存 印刷 リプリント

気象庁によると、高知県四万十市で12日午後1時42分に41.0度を観測した。2007年8月16日に埼玉県熊谷市と岐阜県多治見市で観測した40.9度を抜き、国内の観測史上最高を更新した。

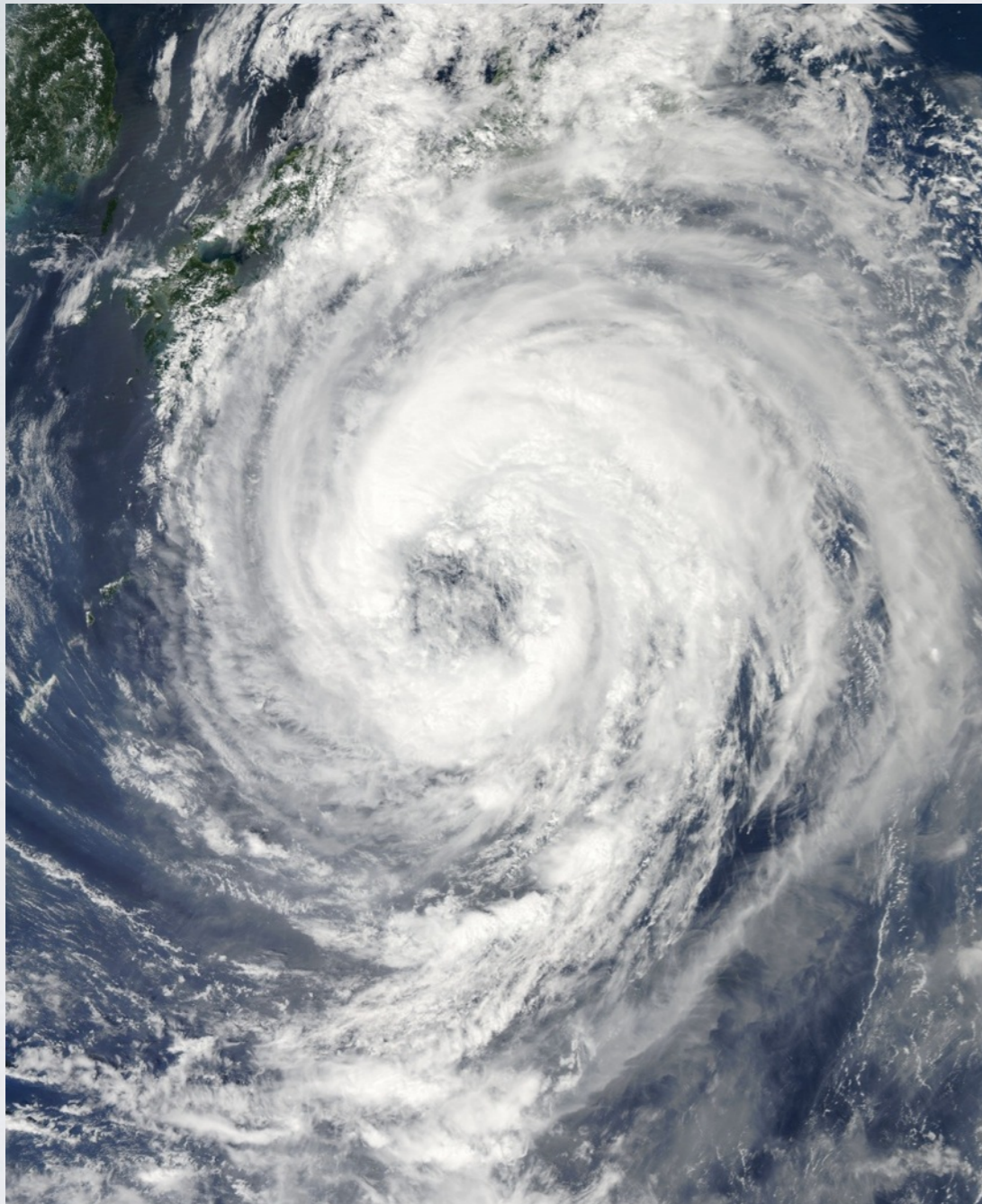
© 2013 Nikkei Inc.



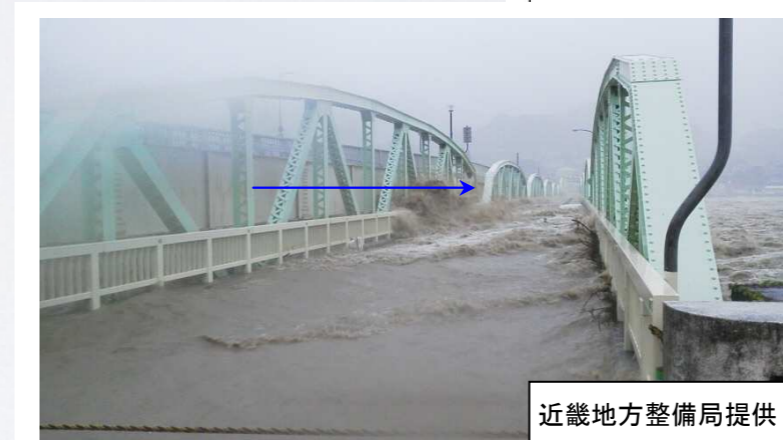
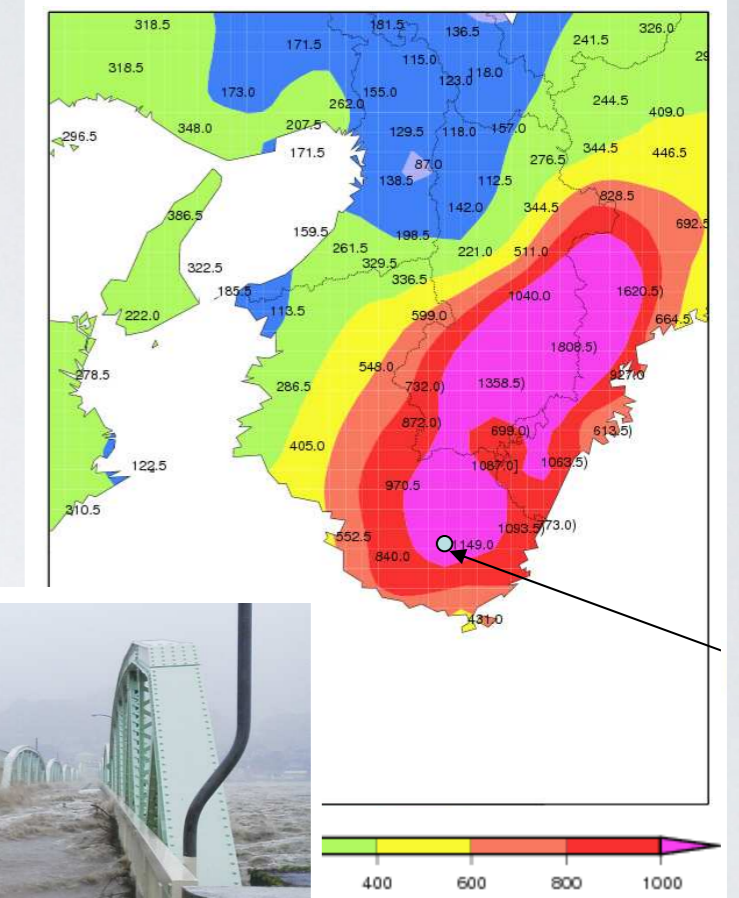
© 2013 The Sankei Shimbun & Sankei Digital

* 日本の観測史上最高気温を記録 (2013.8)

身近にある気象現象



平成23年台風第12号 (Talas)

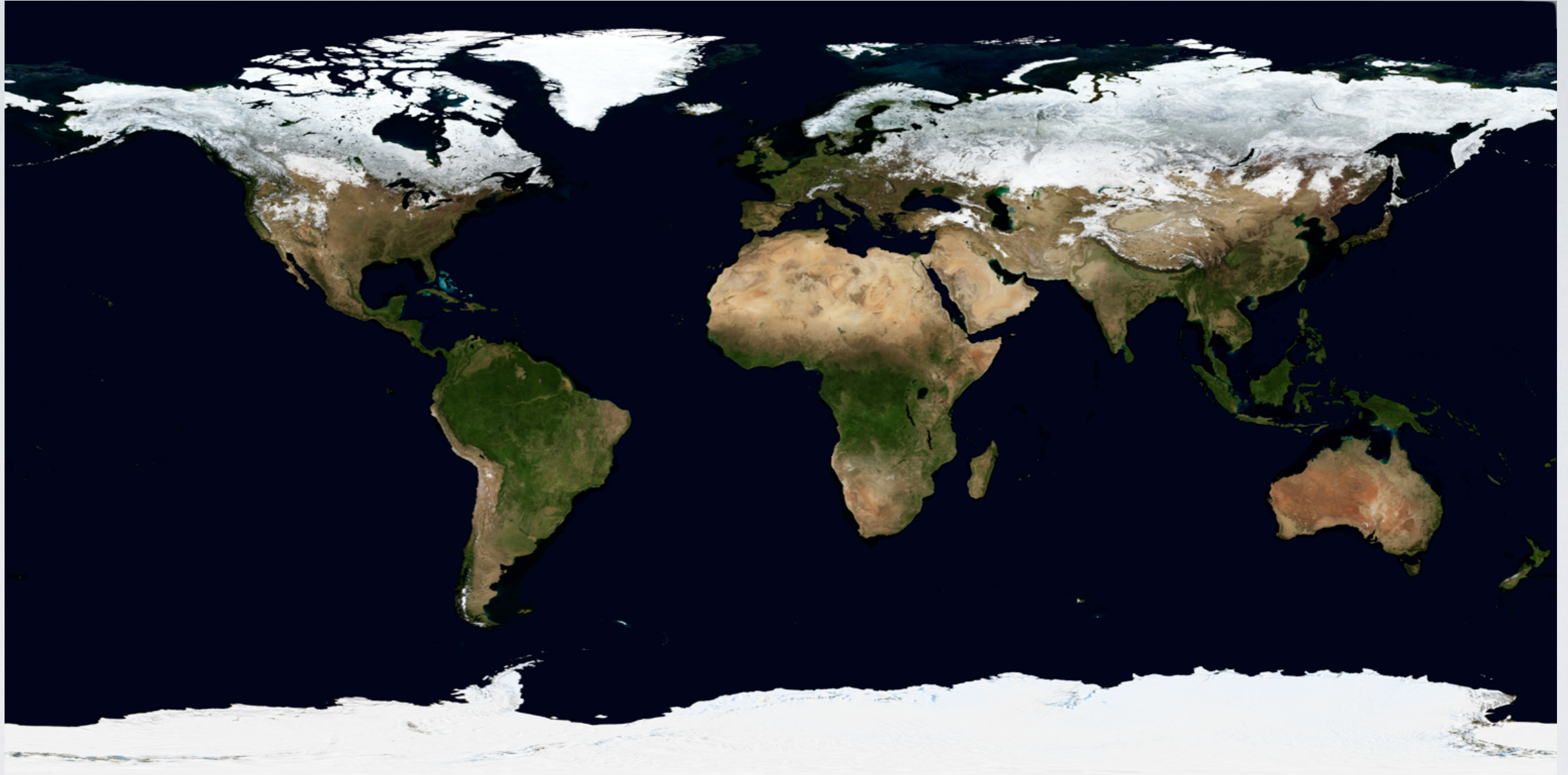


洪水の状況
(熊野川:新宮市旧熊野大橋)

和歌山県 県土整備部 被害復旧記録より

* 2011年台風12号による被害：紀伊半島大水害

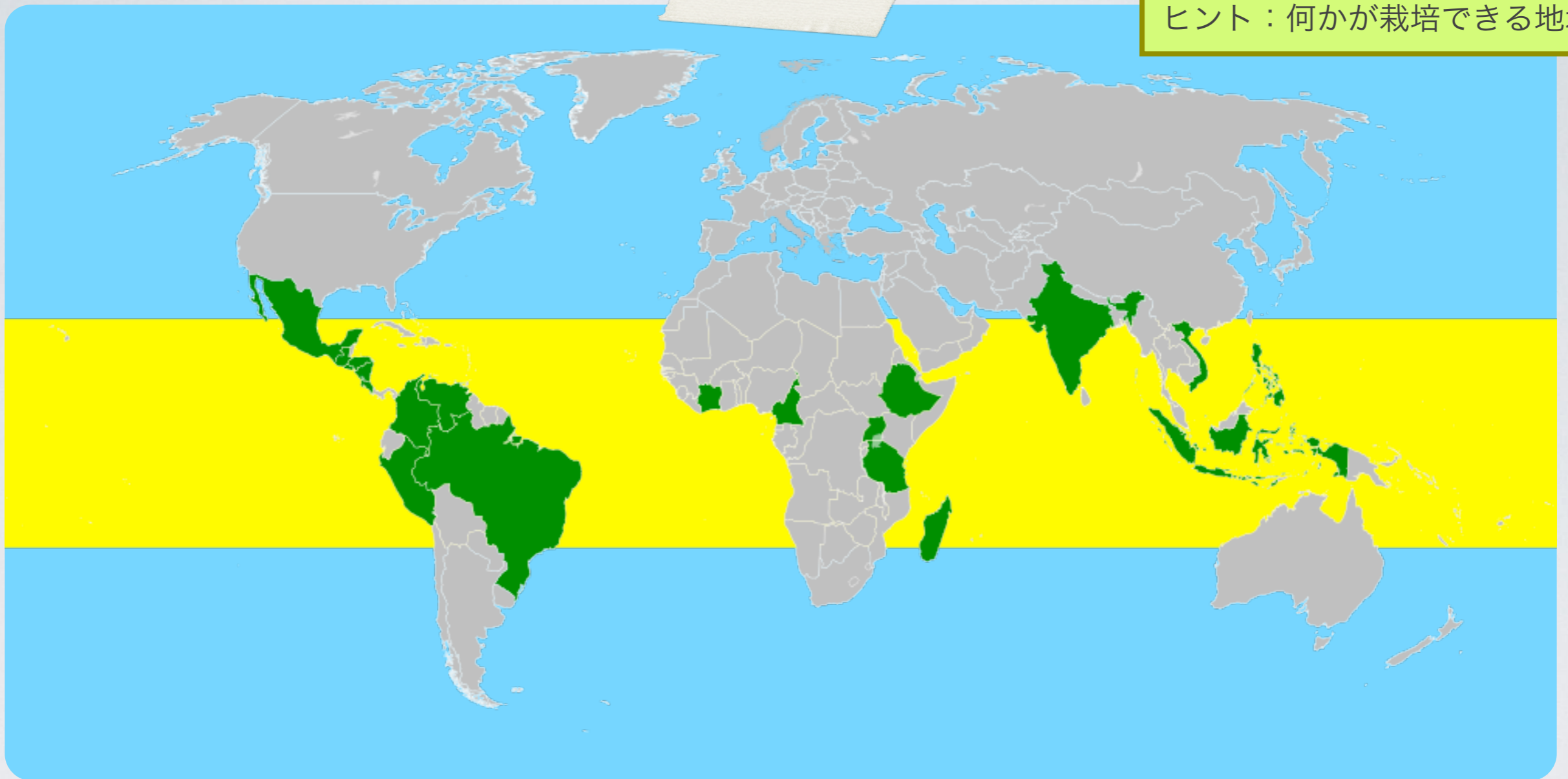
気候と人間



気候と陸上生物圏、人間活動は強く結びついている

気候と人間

黄色い帯は何を表している？
ヒント：何かが栽培できる地域

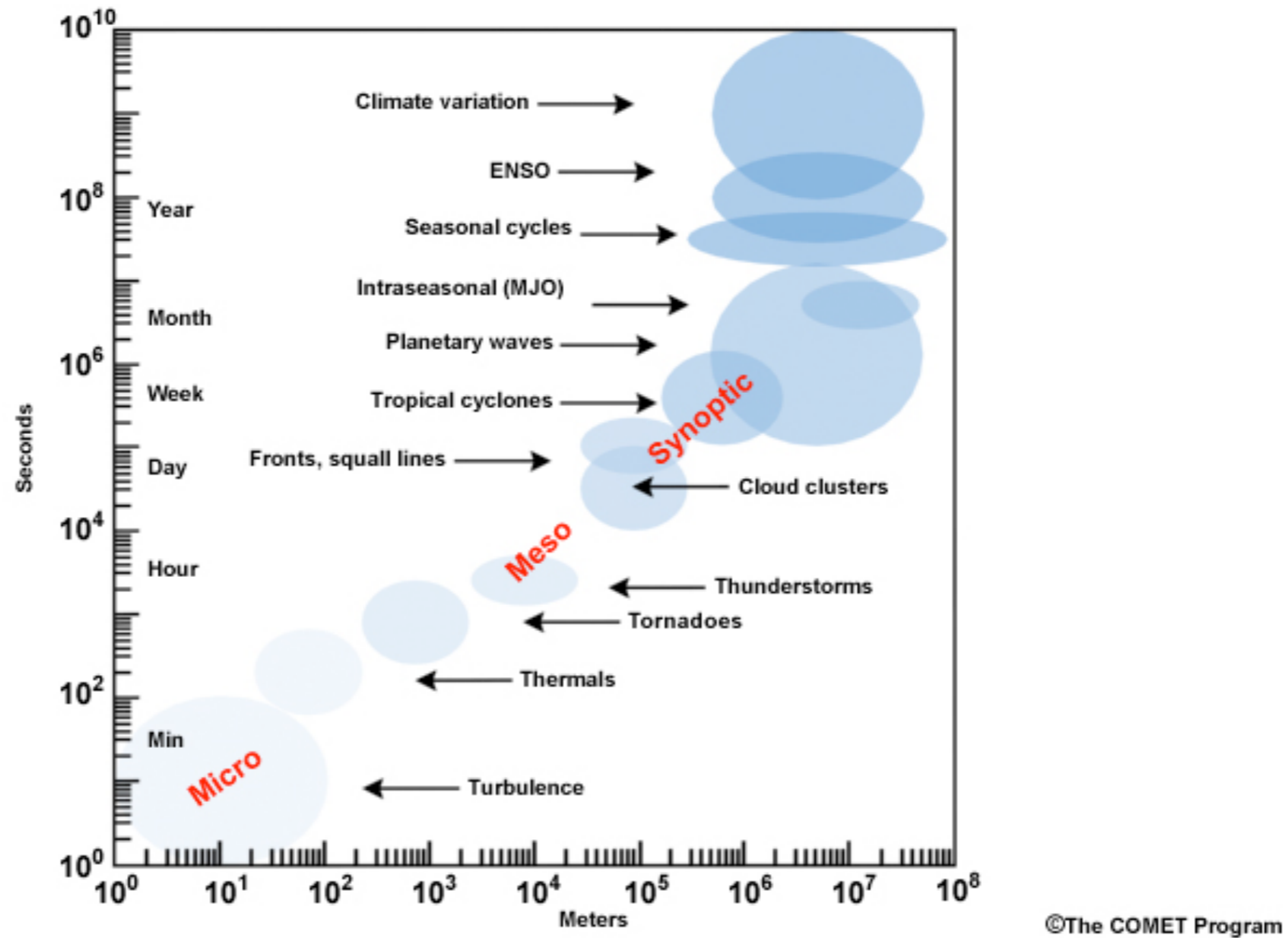


- 気候の変動は何を変えるか？
 - 生態系分布、生物多様性、人間活動（農林水産業、文化、経済）、etc..

大気現象のスケール

- 空間スケールのレンジ
 - 赤道一周の長さは40000km → $O(10^8)$ m
 - 大気現象の最小スケールは？
 - 分子運動、水の相変化：0.1-3nm → $O(10^{-10})$ m
 - 自分の背の高さほどの現象だとしても、1-10m
- 時間スケールのレンジ
 - 季節の移り変わり：数ヶ月～1年
 - 日々の天気の移り変わり：1日～1週間
 - 局地的な降水、日射量、風向の変化：数時間～1日
 - つむじ風、そよ風：数秒～数分

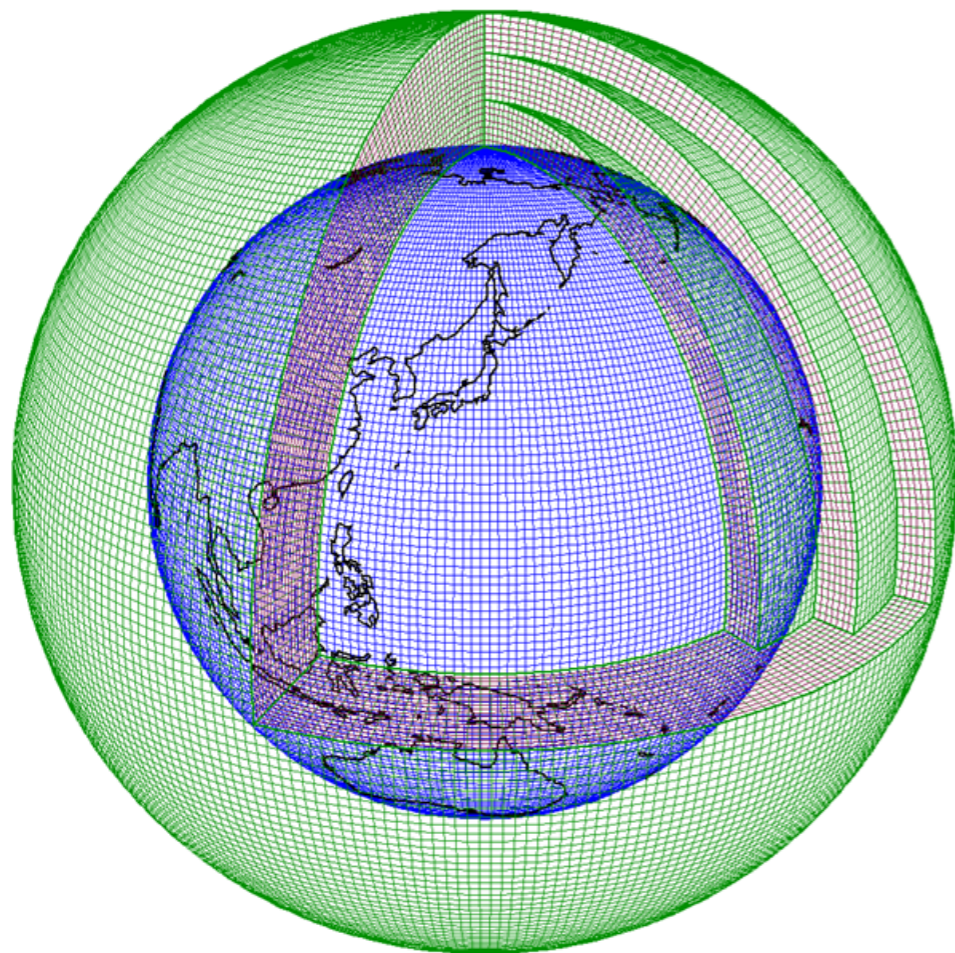
大気現象のスケール



- スケールダイアグラム：各時間=空間スケールの現象が一直線に並ぶ

気象モデルシミュレーションの基礎

数値気象モデルシミュレーション



- 地球大気を格子で覆い、それぞれの格子で気象要素を計算する
- 用途による種類
 - 全球モデル：地球大気全部
 - 領域モデル：一部地域の大気を対象に
 - その他：台風モデル等

数値気象モデルシミュレーション

- 気象分野では、ほとんどの場合モデル=アプリケーション
 - メッシュサイズでどういうモデル化をするか、モデル要素を組み合わせるかが決まってしまう
 - メッシュサイズはほぼ、用途と計算機的能力で決まる
- 流体運動の記述だけでは足りない
 - 解像度の問題：メッシュサイズが大きければ、小さいスケールの流体现象を表現できない→乱流混合モデルや積雲対流モデルとして考慮
 - 湿潤過程の問題：水蒸気・水・氷が相変化しながらエネルギーを運ぶ
 - 境界条件の問題：宇宙(太陽光・赤外線)、海洋、陸面とエネルギーを交換する
 - 初期条件の問題：完全な初期状態を得ることはほとんど困難→観測値や統計を用いて最適な値を作成する

数値気象モデル

- 力学過程：流体力学を解く部分。基礎方程式系はモデルによって異なる

- 連続の式(密度 ρ , 風速 $\mathbf{u}=(u,v,w)$)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

- 運動量の式 (密度 ρ , 風速 \mathbf{u} , 気圧 p)

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot u) = -\frac{\partial p}{\partial x} - \rho f v$$

$$\frac{\partial \rho v}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot v) = -\frac{\partial p}{\partial y} - \rho f u$$

$$\frac{\partial \rho w}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot w) = -\frac{\partial p}{\partial z} - \rho g$$

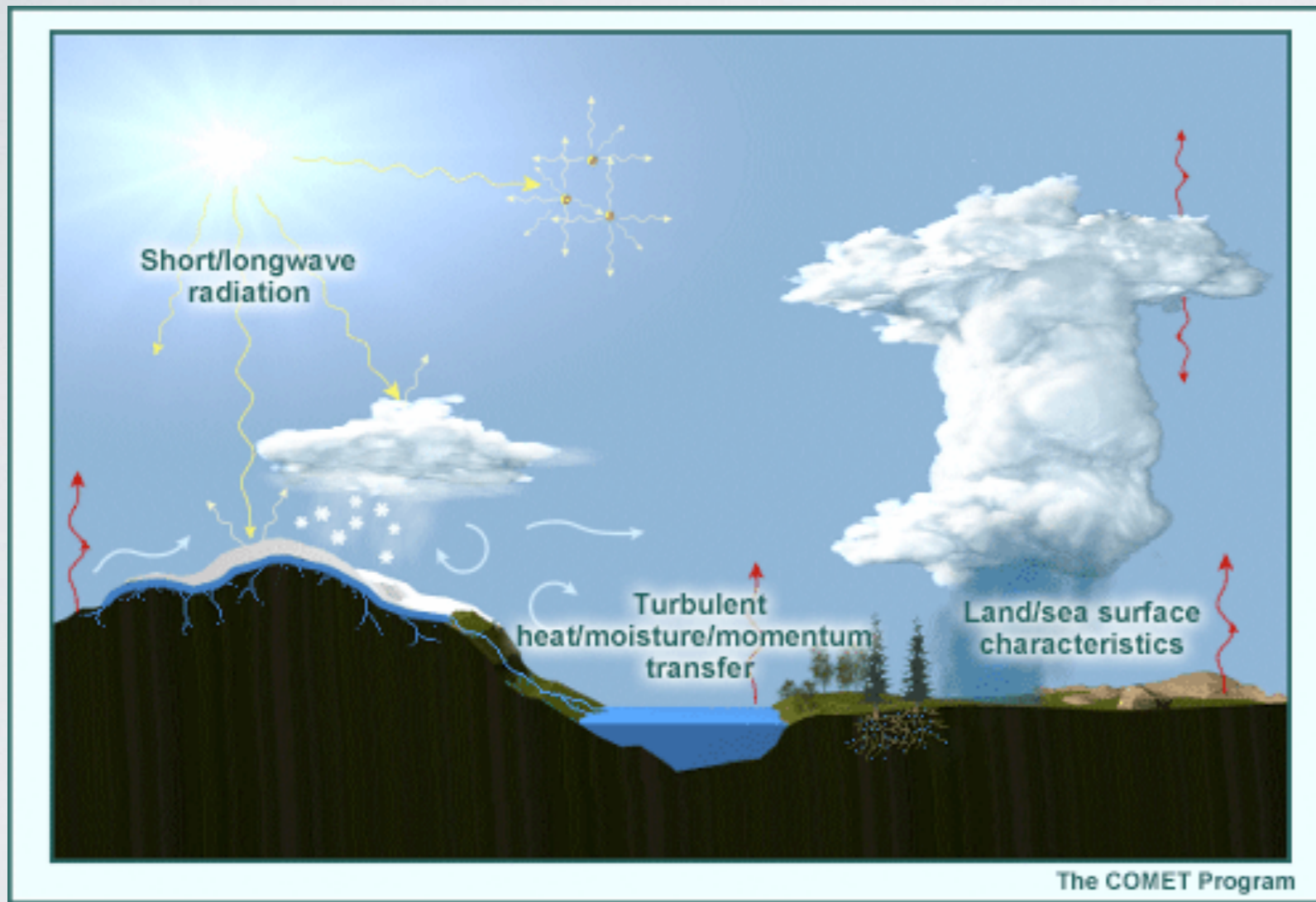
- 気体の状態方程式 (密度 ρ , 温度 T , 気圧 p)

$$p = \rho R T$$

- 熱力学方程式 (密度 ρ , 温度 T , 気圧 p)

$$Q = C_v \frac{dT}{dt} + p \frac{d\alpha}{dt}$$

数値気象モデル



- 諸物理過程を解く部分
 - 雲過程
 - 積雲対流過程
 - 放射過程
 - 地表面過程
 - 乱流混合過程 etc...

モデルの離散化

- 力学過程の離散化

- これまでの主流：スペクトル法

- 球面調和関数で空間場を波の成分に変換して計算
 - 精度が高く、球面上の格子をうまく扱うことができる
 - ルジャンドル変換は波数に対して $O(N^3)$ で計算量が増加
 - 全対全通信が発生するため、計算ノード数が増えると不利

- 近年は格子法が復権する

- 有限差分法、有限体積法、有限要素法、スペクトル要素法など
 - 極問題の回避：両極で格子が集中しない配置の必要性
 - 袖領域の隣接通信で済むので、計算ノードが増えた時に有利

- 移流項の取り扱い

- ：あまり高次の移流スキームは用いられない傾向にある

モデルの離散化

- 物理過程の離散化

- 基本的に粗視化、単純化されている
- 原理的に一意に決まらないパラメータがしばしば存在する
 - ：チューニング次第で結果が変わる
- 演算量は比較的多め：要求するB/F比が小さい
- ループボディが大きく条件分岐が多い
- 水平方向に依存関係を持たない
 - ：水平方向にプロセス並列した場合は、通信が不要

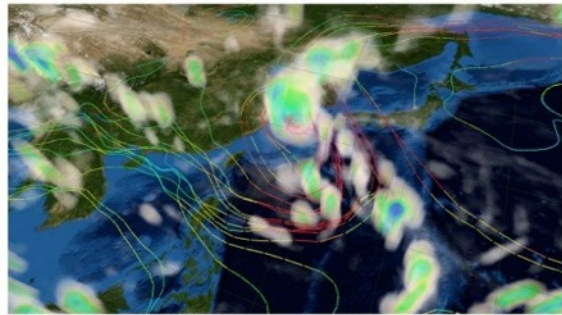
- Byte/FLOPS(B/F)比：演算のためのデータを転送する量と演算量の比

- データ転送は**メインメモリーチップ間**、キャッシュメモリ-演算器間、CPU-GPU間、ネットワーク通信、ファイルI/Oなど様々な部分で発生する

数値気象シミュレーション

Typhoon Bolaven 2012

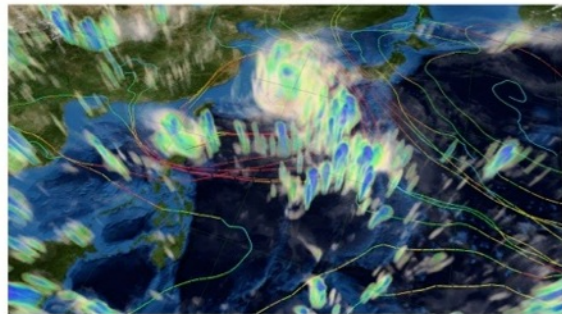
G-level 8



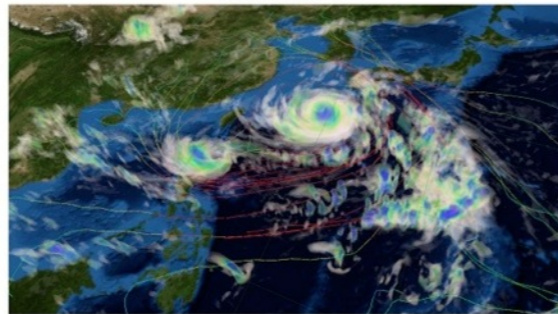
G-level 9



G-level 10



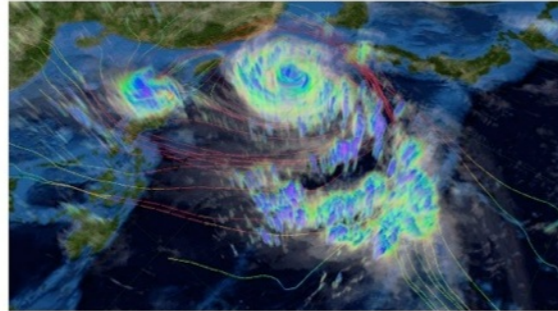
G-level 11



G-level 12



G-level 13



- 気象モデルは「まだまだ足りない」
 - もっと細かくしたい
 - ：雲の細かい構造や、複雑な地形を表現する必要性
 - 粗視化を行わないで計算したい
 - ：諸物理過程で用いているモデルのパラメータに依存する部分が多い
- ➔ 高解像度化で計算量は爆発的に増える：
水平解像度を倍にすると計算量は8倍

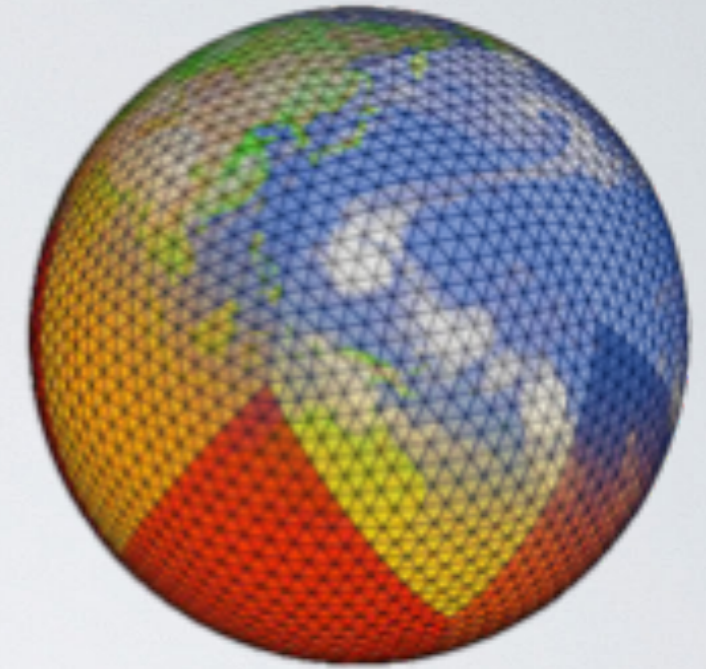
いかに速く計算するか？

- 繰り返し1回あたりに進むシミュレーション時間を延ばす
 - 数値計算では、値が安定して計算できる時間間隔に制限があることが多い
 - 有限体積法のような手法では、格子の幅を細かくとれば時間ステップの刻み幅は小さくせざるを得ない (CFL条件)
 - 高解像度にするほど、時間はかかる
- 繰り返し1回あたりの実行速度を上げる
 - 計算する項目を粗視化して減らす
 - 再現精度は良くなるとは限らない
 - **コンピュータに速く計算してもらおう**
 - 速いコンピュータだけでなく、速く動くプログラムも必要

京における

全球雲解像モデルNICAMの性能評価

全球雲解像モデルNICAM



- **Non-hydrostatic Icosahedral Atmospheric Model (NICAM)**

- 2000年より開発を開始

Tomita and Satoh (2005, Fluid Dyn. Res.), Satoh et al. (2008, J. Comp. Phys.)

- 2004年に地球シミュレータを用いた世界初の全球3.5kmメッシュ計算に成功

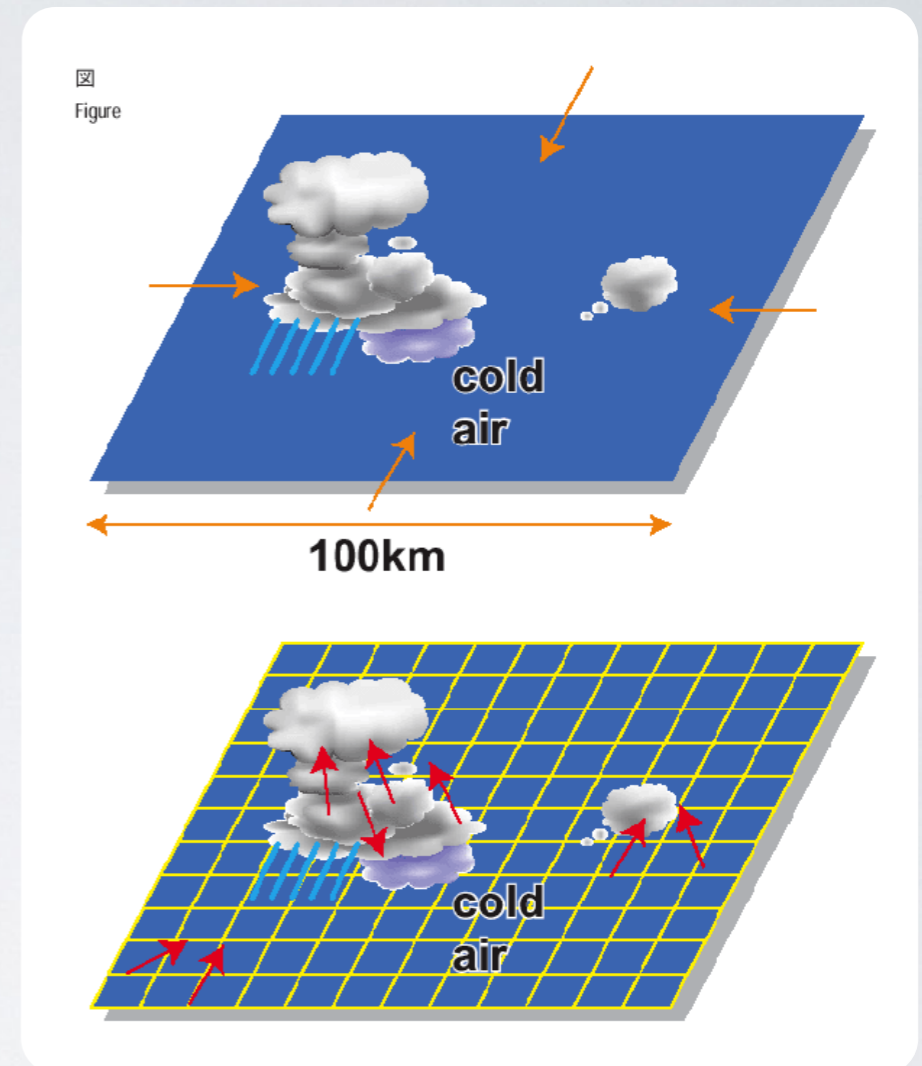
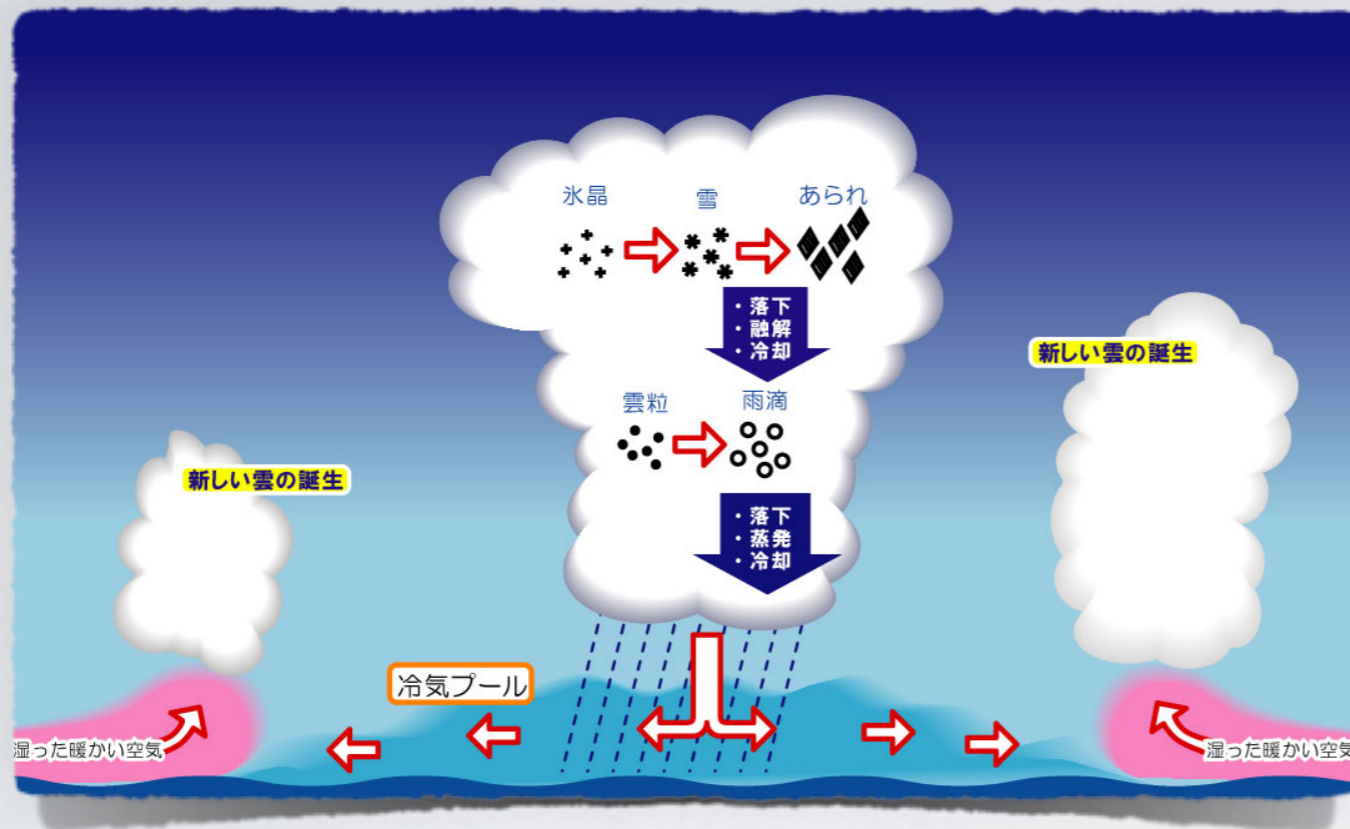
Tomita et al. (2005, Geophys. Res. Lett.), Miura et al. (2007, Science)

- メイン開発者

： 海洋研究開発機構、東京大学大気海洋研究所、理研計算科学研究機構

- Athena project (2009-10)： 欧米の研究機関と連携した大規模計算も

全球雲解像モデルNICAM



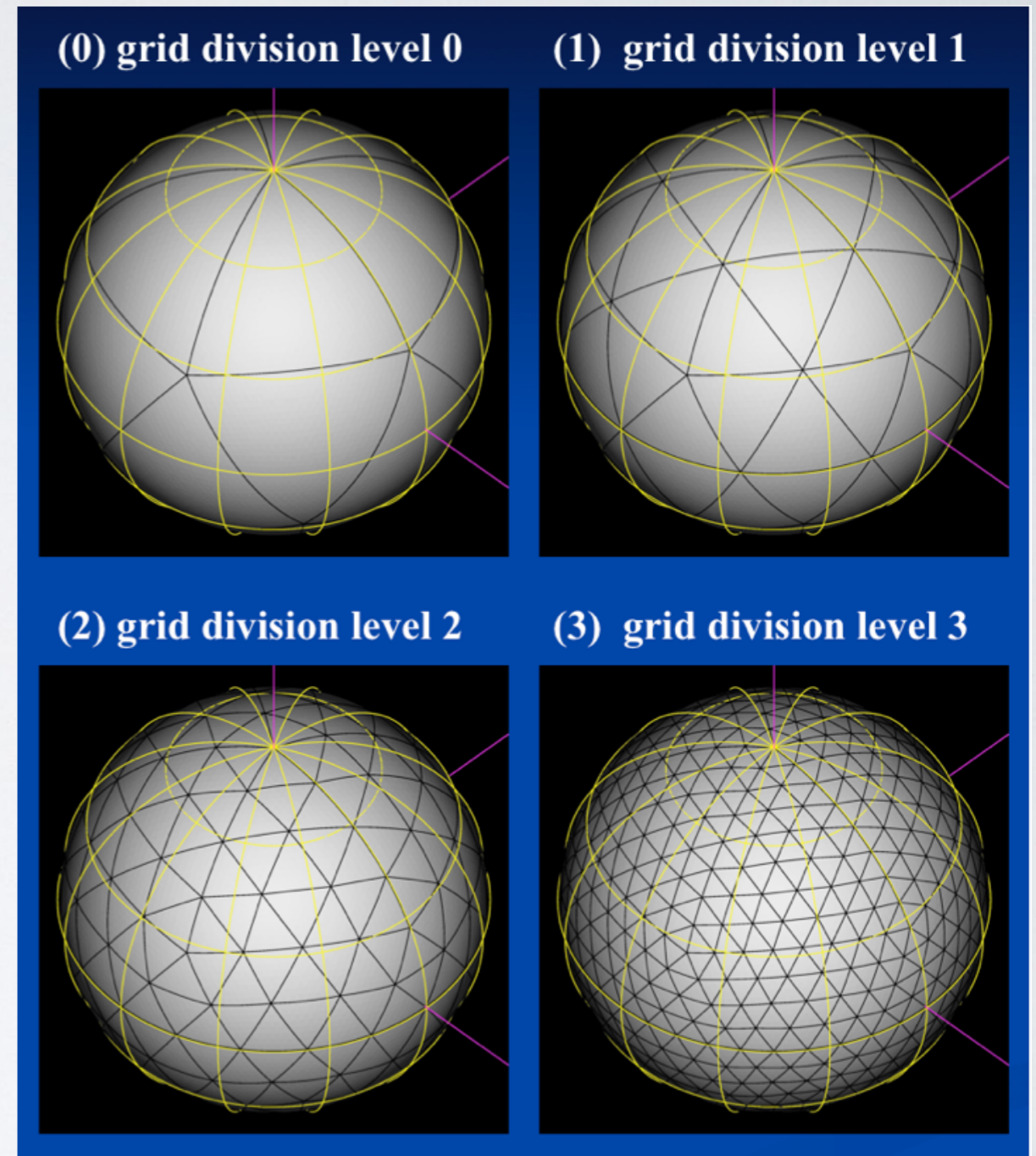
- NICAMのメインターゲット：熱帯の対流システム
 - 中緯度とメカニズムの異なる現象：積乱雲を陽に表現する必要性
 - 積乱雲の水平スケールは~1km：超高解像度シミュレーションが必要

NICAM Framework

- 正20面体格子系
 - 球面準一様格子：等方性に優れる
 - バネ力学による平滑化
 - ：更に一様性を上げる工夫

(Tomita et al. 2001, 2002)

 - Shmidt変換による格子の集中
(ストレッチ)も可(Tomita et al. 2008)
-
- 高並列計算機の利用を想定した設計
 - ：ノード間通信、I/O



NICAMの力学過程

支配方程式系	完全圧縮非静力学方程式系
空間離散化 水平グリッド 鉛直グリッド 地形の扱い	有限体積法 正20面体格子+ばね平滑化 (Arakawa A) Lorentz格子 Terrain-following
保存性	総質量・総エネルギー保存 (Satoh 2002, 2003)
時間離散化	遅いモード：陽解法 (ルンゲ・クッタ) 速いモード：HEVI
移流スキーム	Miura(2007), CWC: Niwa et al.(2011)
数値フィルター	4次の超粘性、4次のDivergence Damping

NICAMの物理過程

物理過程	デフォルト	オプション (青字は導入中)
積雲		Prognostic A-S, Chikira, kuo, tiedtke,
雲微物理	NSW6	LSC, Kessler, G98, lin, WSM6, NDW6
放射	mstrnX	
境界層乱流	MYNN2	MY2, MY2.5, MYNN2.5, MYNN3, Smagorinsky
陸面	MATSIRO	bucket
海洋	slab ocean	COCO
湖・河川		TRIP
地表面	Uno95	
エアロゾル		SPRINTARS
大気化学		CHASER

(各スキームのリファレンスは論文参照のこと)

NICAMのフレームワーク

- 並列化手法

- プロセス並列：MPI並列（水平方向に格子を分割）
- スレッド並列：京では自動並列を利用。OpenMPは未実装

- データ構造

- 構造格子
- (i,j,k,l) =(水平グリッド,鉛直グリッド,水平格子グループの単位)
 - ベクトル機で性能が出せるよう、最内のループ回転数が一番大きい順番に

京コンピュータへの移植

- 地球シミュレータ（並列ベクトル機）から京コンピュータ（超並列スカラー機）へ
 - 最初の時点での実行効率：京のピーク性能比4%
 - AICS運用技術部が中心となった最適化
 - ：力学過程での収束・発散・勾配計算部分等、重要だと判断されたサブルーチン・計算ループ部分（＝演算カーネル）はそれぞれピーク比10%以上にまで高速化
- しかし。。。アプリケーション全体での高速化：4%→5%に留まる
- なぜ？
 - 抽出して最適化した各カーネルは演算量のランキングで上位ではあるが、そもそも全体に占める割合が少ない
 - それらが高速化して、さらに割合が減った→他の部分が問題に
 - ：一般的にアムダールの法則と言われる

京コンピュータへの移植

- この時点で多く用いられた最適化の手法
- 並列性を上げる努力
 - まずはメモリ転送性能を最大限まで活用するだけの、効率的な命令発行数や高い演算密度が達成されていなければいけない＝CPUを遊ばせない
 - 具体的には、コンパイル時にSIMD化やソフトウェアパイプライン化、スレッド並列化がちゃんと適用されるようにする
 - ループ構造の見直し
 - ループの交換
 - ループの融合、分離 等
- キャッシュチューニング
 - メモリ転送性能を節約するため、一度の読み込みで「ひとかたまり」として読み出されるデータが効率よく使われるようにする
 - 同時に演算に使われる小さいサイズの配列次元を一番前にもってくる

なぜ最適化が難しいのか

- “Flat profile”
 - 演算量のホットスポットがほとんど存在しない
 - 最適化はモグラたたきになる：全ての部分で計算時間を減らす努力が必要
- CPUからHDDまで、どの階層でもデータ転送性能を要求
 - 高い要求B/F比
 - 通信も少なくない
 - 頻繁なファイルI/O
- ソースコード規模が大きく複雑
 - 数万から数十万行
 - それぞれの分野の研究者が開発したモデルの複合体

人海戦術

- **Detailed check of performance**

- 数百のFLOPカウンタ、タイマーを挿入
- "time-wasting"領域をリストアップ
 - 計算量が少なく、時間だけかかる部分
 - ：配列のコピー、不必要な配列のゼロ埋め、不必要な中間配列の作成
 - 並列化の妨げになる部分
 - ：複雑なループ構造、最内ループに存在する条件分岐

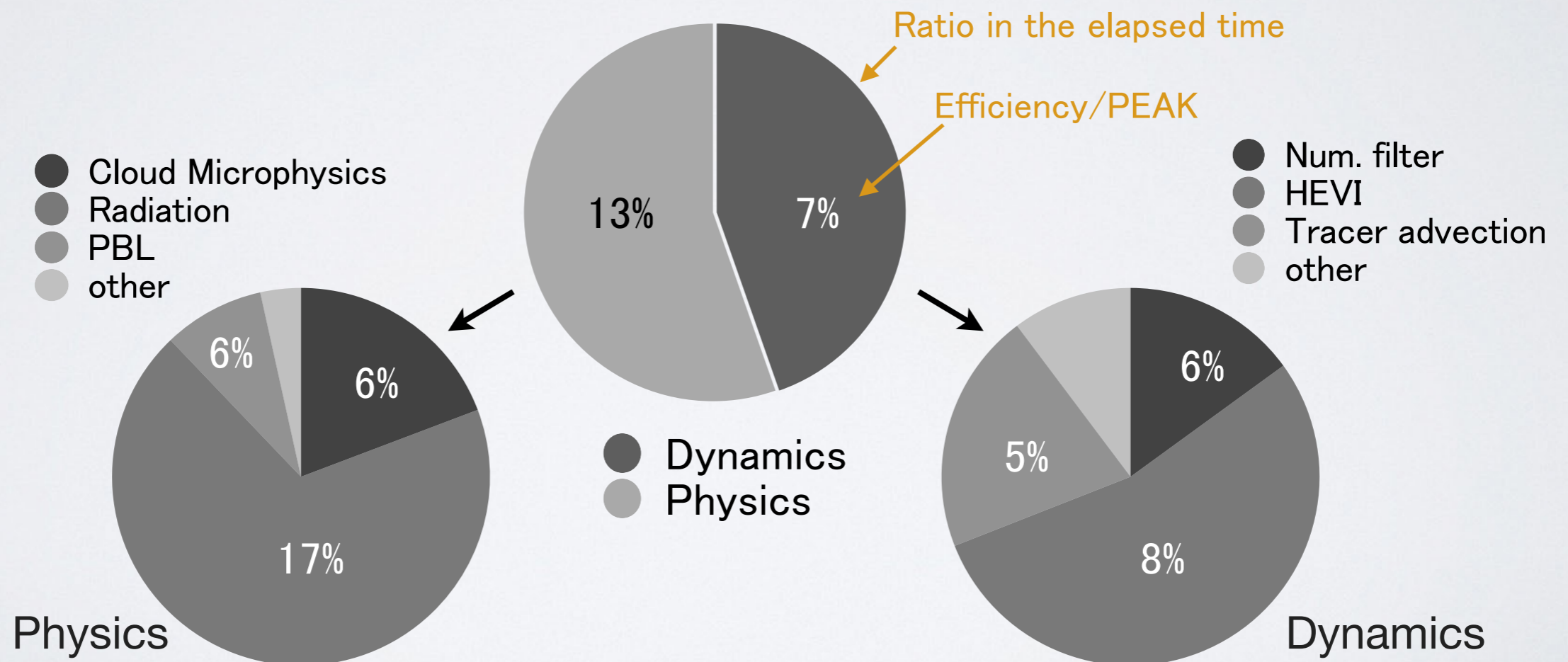
➡ Reduce intermediate arrays, avoid unnecessary zero-filling, and turn the conditional branches out of loops

- Our lesson: the compiler is not so intelligent as we have expected.
 - Compiler-friendly code and readable code are compatible: simple is **best**

Efficiency of NICAM on K Computer

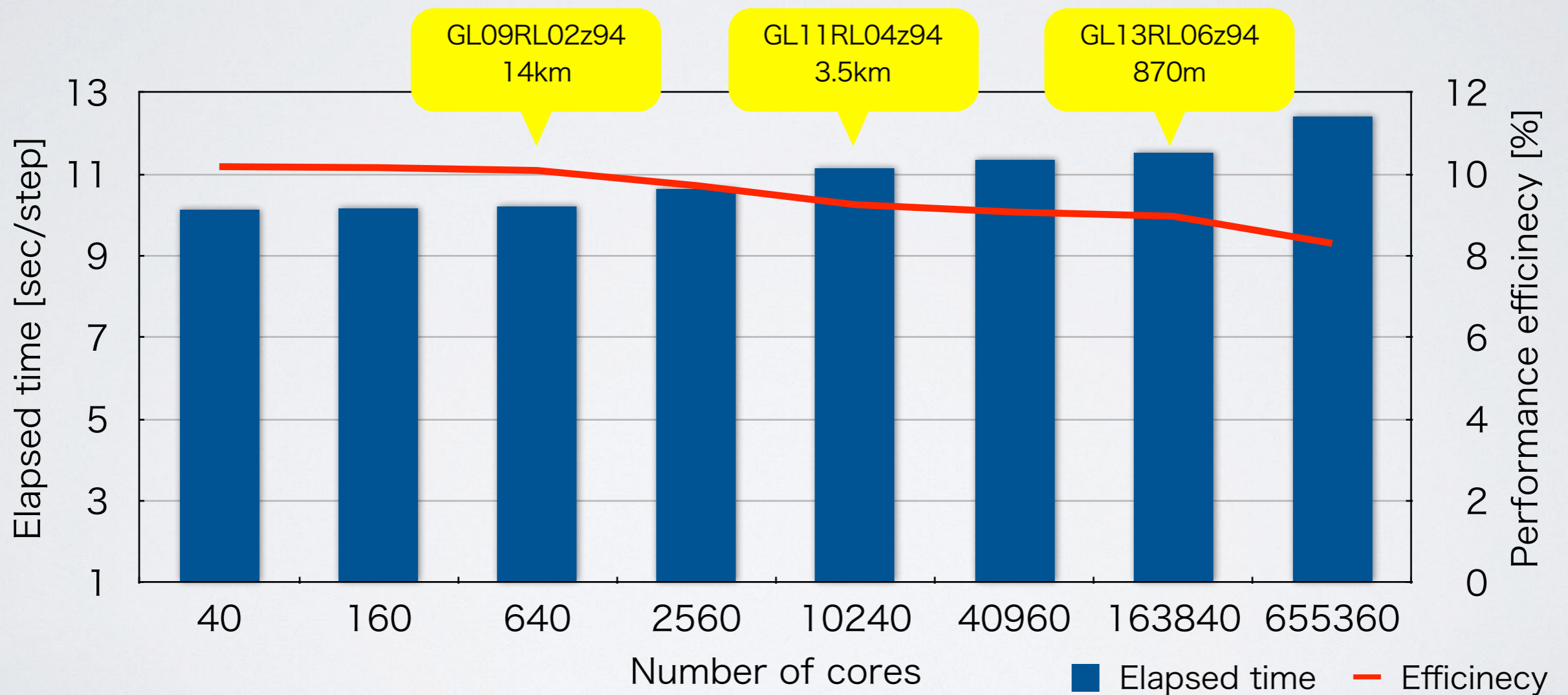
- **Performance efficiency**

- Just after porting from Earth Simulator : ~4%
- Cache optimization to stencil operators : ~5%
- Cleaning the time-wasting codes : ~7%
- Modify conditional branches, refactoring : ~10%



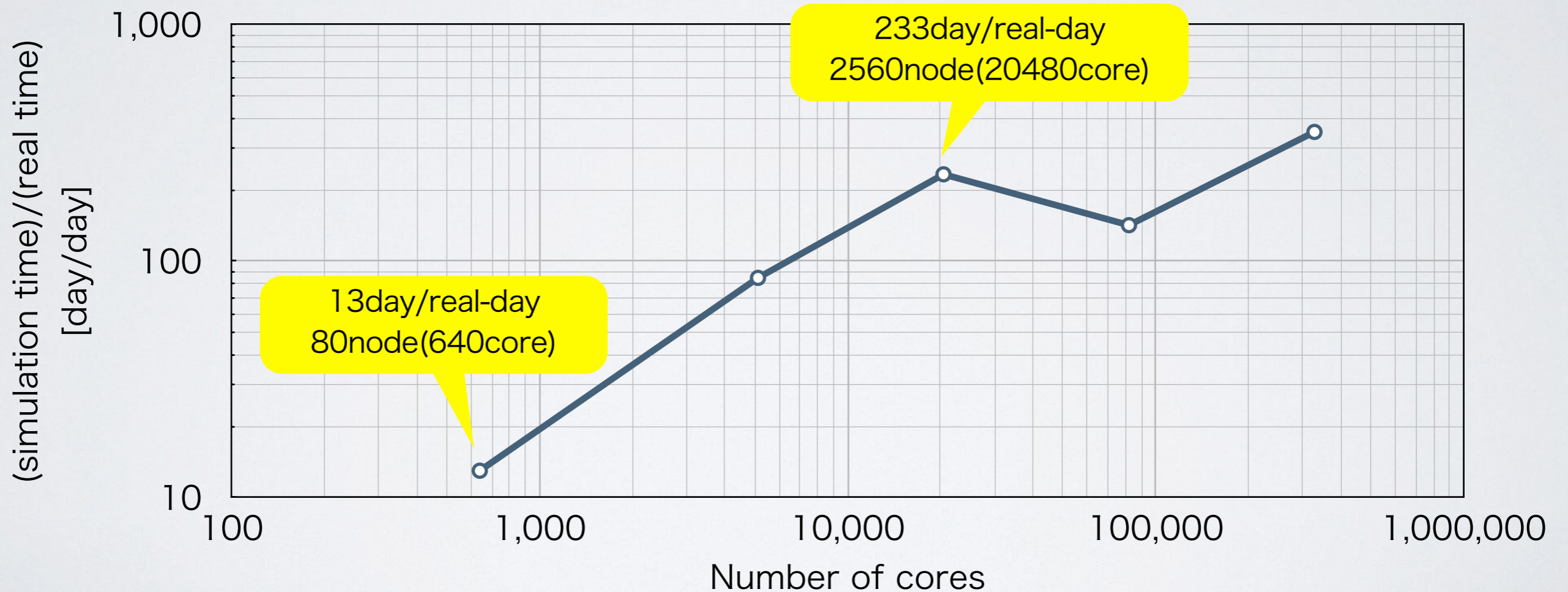
Weak scaling test

- Same problem size per node, same steps
 - Full configuration / full components
 - Realistic boundary / initial data set
- Good scalability up to 81920node x 8threads with **0.9PFLOPS**



Strong scaling test

- 14km horizontal, 38layers, total problem size is fixed
- The efficiency decreases rapidly
: the relative ratio of the communication time increases



Global Sub-km experiment

・ 実験設定

- ・ 水平解像度：870m、鉛直層数：40kmまで94層→総格子数：660億
- ・ (最終的な)積分時間：48hour、 Δt ：2sec→総ステップ数：86400step
- ・ 使用ノード数：20480ノード（163840core）
→演算性能：210TFLOPS（ピーク性能比8%）
- ・ 実験にかかった時間：1時間の積分に4.5時間
- ・ メモリ利用量：200TByte、総出力データ量：320TByte（リスタートを除く）

- ・ ポストプロセス、プリプロセスにも膨大な時間が
：解析も京で行う必要がある

- ・ ちなみに最近の身近なマシンだと・・・
 - ・ Mac Book Pro：86GFLOP→6%で動かして0.005TFLOPS
 - ・ 32core Linuxクラスタ：700GFLOPS→6%で動かして0.04TFLOPS

今後に向けた取り組み

京での最適化で得られた経験

- その分野の科学者(domain scientist)は計算機のことを知るべきだし、計算機科学者はその分野のアプリケーション特性（文化も含めて）を知るべき
 - ：それぞれの専門家が最大限の能力を発揮するには、お互いの理解を深める必要がある
- 大規模計算の世界はFIのようなもの
 - 如何に計算機を意識してコーディングするかが重要
 - FORTRAN77最速説：オブジェクト指向等、コンパイラがうまく解釈して最適化してくれない部分はまだある
 - ファイル出力の分散並列の徹底
 - シミュレーションの実行を最も邪魔しないのは、非同期・分散
- 1桁違う問題規模に挑戦するたびに、新たな問題が浮上する
 - ：プリプロセスから解析までの研究ライフサイクル全体が超並列化する必要

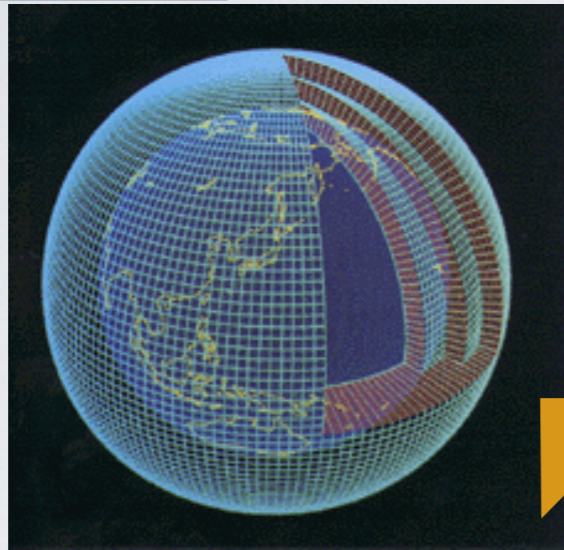
高性能・高生産性の気象モデルをめざして

- 気象・気候モデルは年々コード規模が肥大化している
 - 到底一人でスクラッチから作ることは出来ない
- 計算機アーキテクチャとプログラミングモデルは変化している
 - 追従しなければせっかくの計算機性能を活用できない
- どうする？
 - モデルのAPIを統一し、別のモデルでの開発資産を皆で共有する
 - カプラーによる弱結合のモデル連成実行
 - 複数研究機関による大気モデル共通基盤ライブラリの構築
 - ソースコードをオープンソース化し、様々な研究者が新たに参入しやすい環境をつくる
 - コードを増やす勇気、コードを削る勇気：網羅的なテストの充実が重要

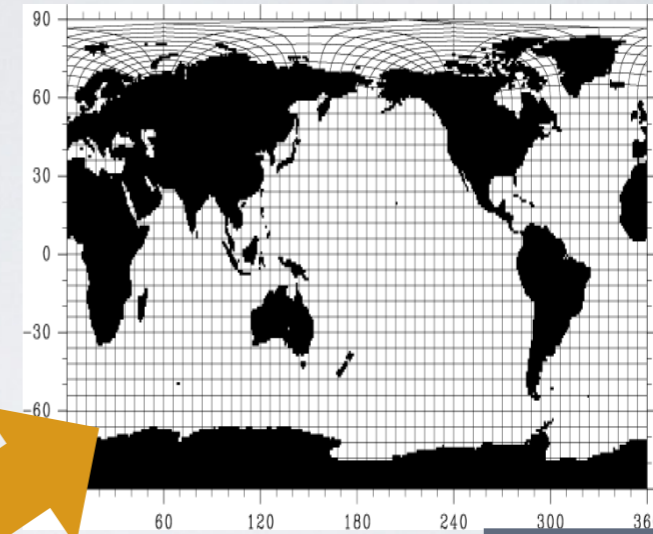
カプラーを用いた異なるモデルの連成計算

MIROC-AGCM

格子系：緯度経度

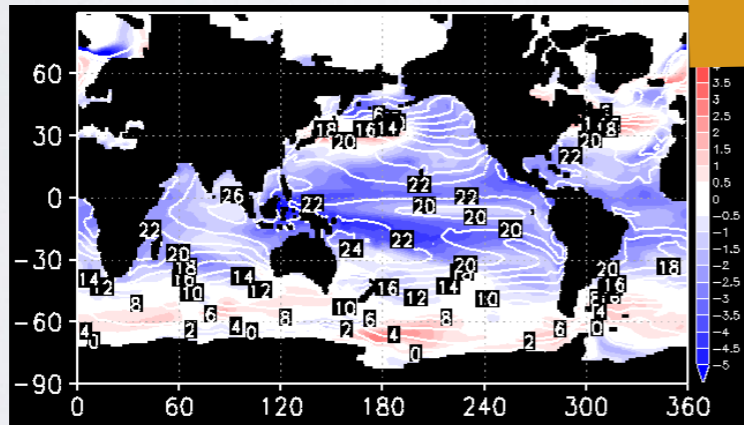


格子系：tri-polar



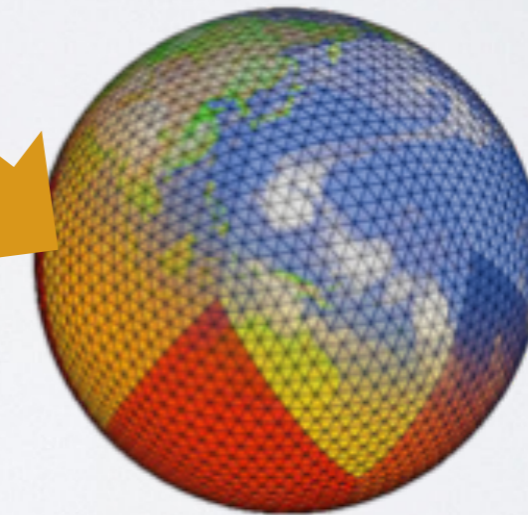
MIROC-OGCM

File IO/Postprocess



格子系：緯度経度

J-cup



NICAM

格子系：正20面体

カプラー (Jcup) はMPMD実行される各サブモデル間の通信管理と格子変換を担う

NICAM Coupled I/O

プリプロセス(並列実行)

正20面体格子配置データ

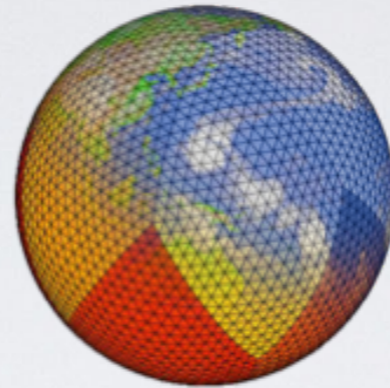
mklmap

指定した緯度経度格子に対する
変換テーブル作成



変換テーブル・係数は分散ファイルで

モデル計算(並列実行)



分割領域をそれぞれ指定ステップ毎に
ファイル出力



ポストプロセス(シリアルまたは並列実行)

ico2ll

格子系の変換、集約



NICAM Coupled I/O

プリプロセス(並列実行)

正20面体格子配置データ

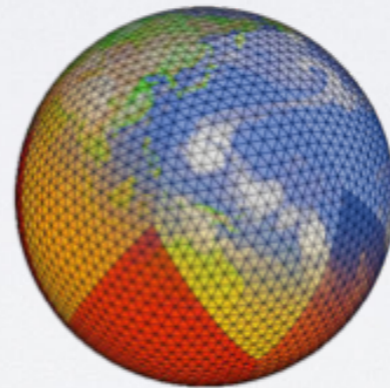
mkllmap

指定した緯度経度格子に対する
変換テーブル作成

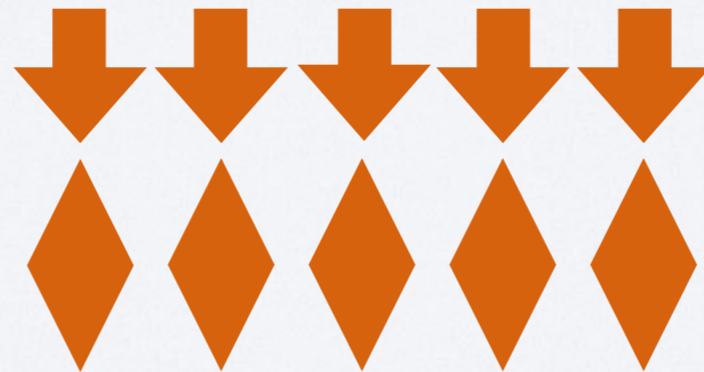


変換テーブル・係数は分散ファイルで

モデル計算(並列実行)



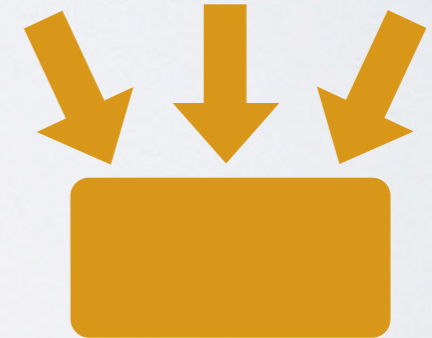
分割領域をそれぞれ指定ステップ毎に
ファイル出力



Coupled IO(並列実行)

Jcup: 通信と格子系の変換

指定ステップ毎に
1ファイルをMPI-IOを用いて出力



大気モデル共通基盤ライブラリの構築

- 気象・気候モデルのための共通基盤ライブラリ環境 facilitator : 西澤誠也(AICS)
- 参加グループ
 - 気象庁・気象研究所
 - MIROC開発チーム (東京大学, JAMSTEC, 国立環境研, 等)
 - NICAM開発チーム (東京大学, JAMSTEC, 理化学研究所, 等)
 - CReSS開発チーム (名古屋大学)
 - AFES開発チーム (JAMSTEC)
 - MSSG開発チーム (JAMSTEC)
 - GAIA開発チーム (九州大学, NICT, 等)
 - Jcup 開発チーム (RIST, 等)
 - MATSIRO 開発チーム (東京大学, 等)
 - 地球流体電脳倶楽部 (北海道大学, 京都大学, 神戸大学, 九州大学, 等)
 - Team SCALE (理化学研究所)
- 現在は物理過程のライブラリ化・相互利用に向けて活動中
： データ構造・APIの決定、評価方法の確立等

SCALEプロジェクト

- 計算科学研究機構において、気象科学の研究チームと計算機科学の研究チームが共同し、新しい気象・気候ライブラリのコデザインを行う



Team SCALE



Computational Climate Science Research Team

TL : H. Tomita



System Software Research Team

TL : Y. Ishikawa



Programming Environment Research Team

TL : M. Sato



Programming Framework Research Team

TL : N. Maruyama



HPC Usability Research Team

TL : T. Maeda



Library, and Research for weather/climate research
With computer science people (SCALE library)

オープンソースへの取り組み



- NICAM-DC：力学コアパッケージ
 - BSD 2-clause ライセンス
 - ウェブサイト (<http://scale.aics.riken.jp/nicamdc/>) とGitHubから入手可能
 - 基本的なテストケースを揃える
- 広がる利用範囲
 - SCALE libraryへの統合
 - 多国間国際連携事業 ICOMEX project：科学的・計算機的な性能評価に利用
 - 将来のスパコンのためのベンチマークソフト、ミニアプリに
 - GPUへの移植

NICAM-DC with OpenACC

- With the **support** of the specialist of NVIDIA Japan (Mr. Naruse)
- **Advantage of OpenACC**
 - : Small modification, directive-based
 - Reduce the time to develop
 - Keep from splitting the source code
 - In the 58,000 line of NICAM-DC source code, only 2000 lines are modified or added
- **Performance evaluation on TSUBAME 2.5 supercomputer**
 - Largest GPU cluster in Japan
 - : 1300+ nodes, 3GPUs per node



OpenACC
Directives for Accelerators

Acceleration by GPU

- Performance test on TSUBAME2.5

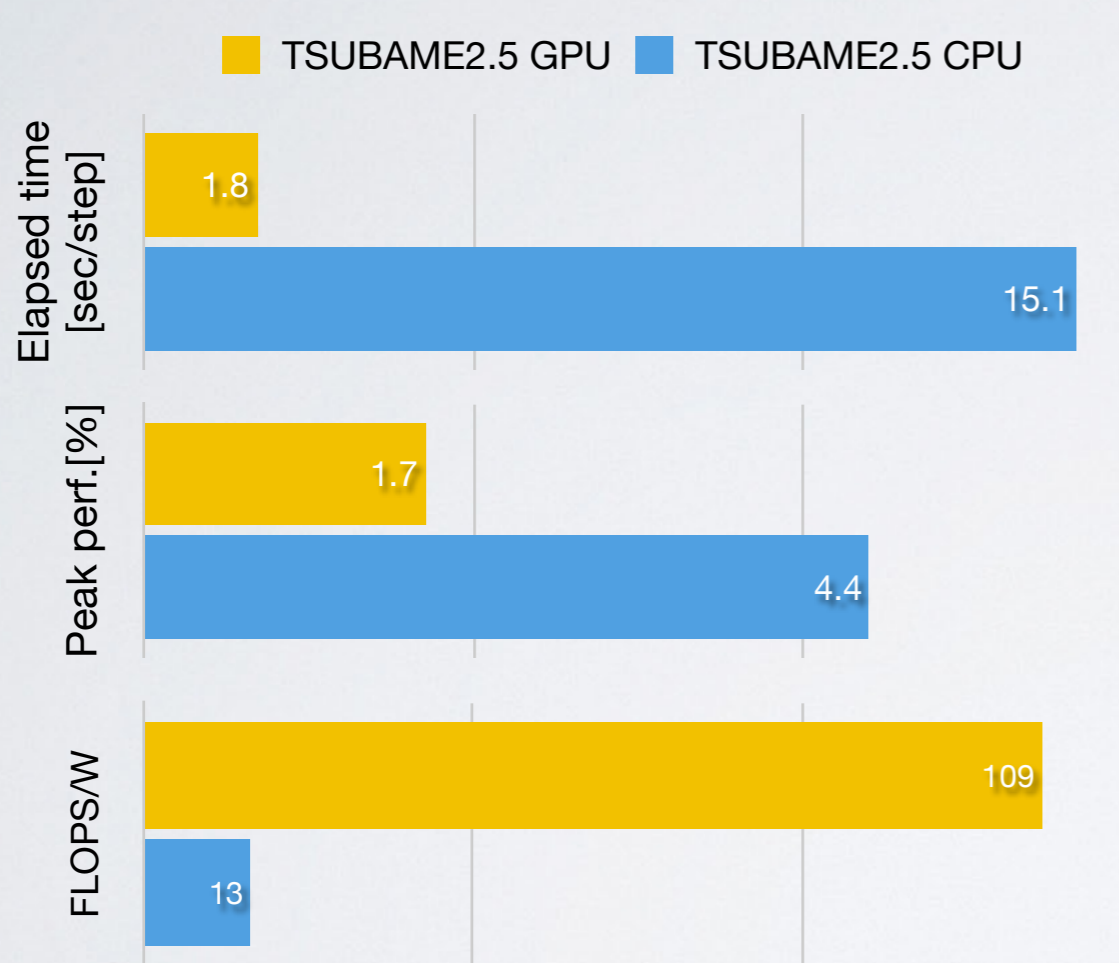


図2: TSUBAME2.5でNICAM-DCを実行した際の実行性能。それぞれメインループ1ステップあたりの所要時間(上)、演算ピーク性能比(中)、ワットあたりFLOPS(下)を示す。

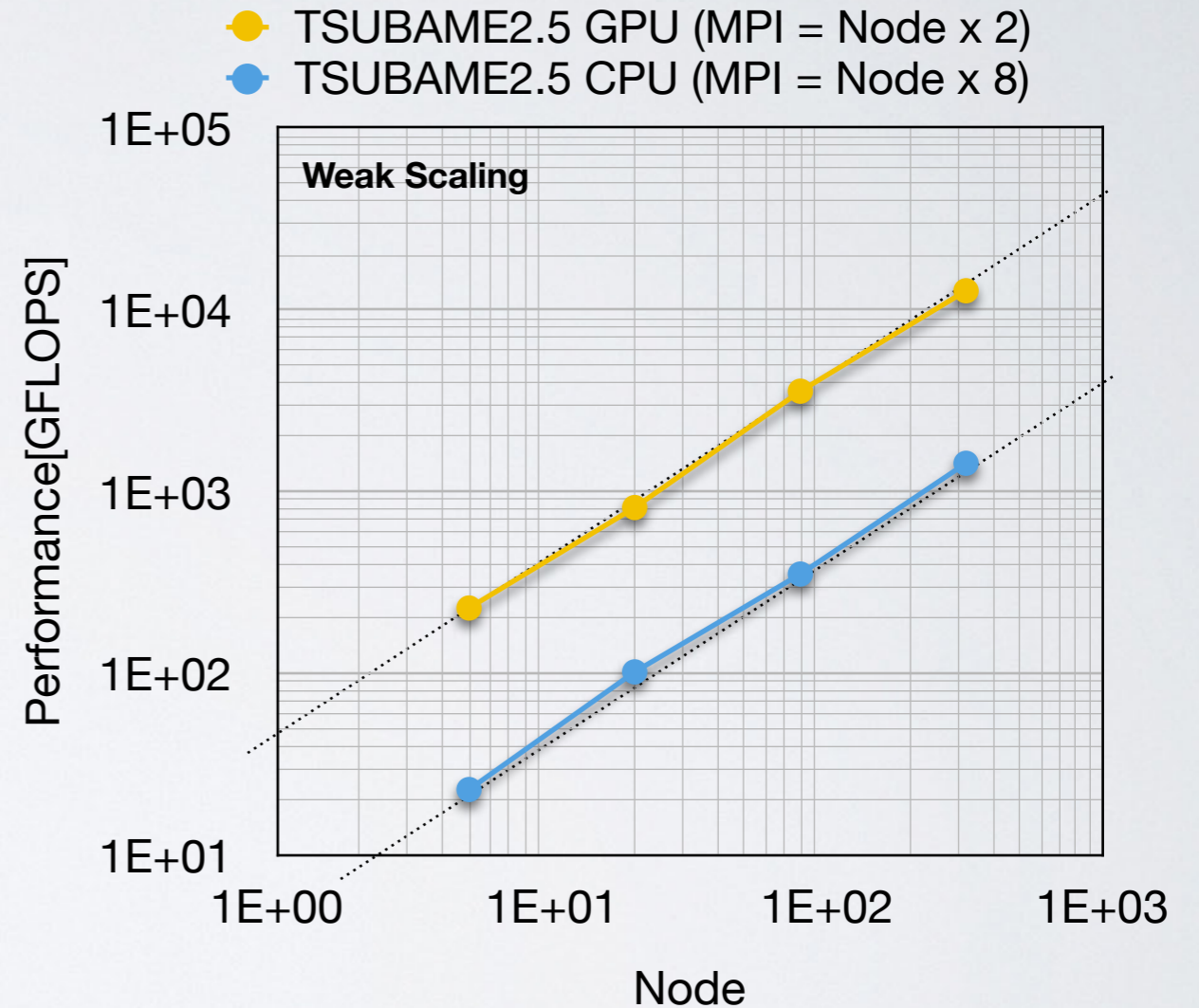


図3: TSUBAME2.5でNICAM-DCを実行した際のウィークスケーリング性能。

最後に

- 技法は1年、ニーズは10年、ヴィジョンは100年スパン
(MITメディアラボ副所長・石井裕教授の言葉)
- 野望をもって、サイエンスをデザインすることの重要性
- 大規模計算の経験は、大規模計算をしなければわからない
 - コードの書き方も変わってくる
 - 複数人が協同作業する体制になって、必要になることもある



ありがとうございました