

ポスト「京」プロジェクトの概要

石川裕

システムソフトウェア研究チームリーダー

理研計算科学研究機構

10:30～11:00

社会的・科学的課題とシステム設計の基本的考え方

- Science-driven System
 - 計算科学ロードマップ(第2版)に基づく概念設計
 - アプリFSにおいて社会的・科学的課題および必要とする計算資源量について精査を行っている
 - ロードマップで見えてくる使い方は
 - 大規模、精密、長時間発展といったcapability computingのニーズ
 - 複雑な現象を対象とした課題におけるensemble computingのニーズ
 - Big data computing、社会科学シミュレーションのニーズ
 - これらの要求に応えるシステムの設計が前提
- Sustainable System
 - 京の後継機として京の資産が受け継げる、そして、将来の計算機システム発展動向を見据えたシステム
 - 2020年ころの世界の最先端システム、競争力、競合力あるシステム
 - 将来の我が国のHPCの発展につながるシステム
- TCO(Total Cost of Ownership)-aware System
 - 白書で必要とされる計算資源量に対して、低電力、ソフトウェア高移植性、高耐故障性を有するシステム設計

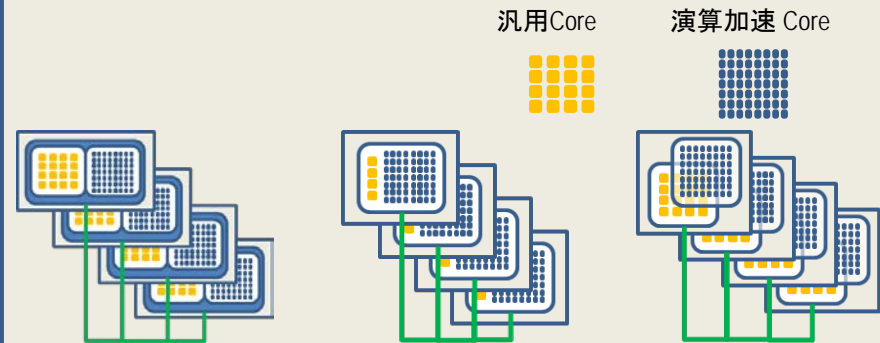
システム設計基本的考え方

- 将来動向
 - 高い性能電力比と不規則非構造データ処理もできるCPUとして、汎用コアと演算加速コアが統合されていく。しかし、どのように統合されていくかは今後の研究成果に依存
 - ノード単体の演算性能向上に比べ搭載可能メモリ容量はさほど増えない
- 開発に関する考え方
 - システム設計
 - 汎用部(汎用コア)と演算加速部(演算加速コア)を有するマシン
 - 汎用コアでないと性能がだせないアプリケーション、演算加速コアで性能だせるアプリケーション、両コアを使って性能がだせるアプリケーションを棲み分け、全電力時間積削減
 - 将来にわたって有効な統一プログラミングモデル、ライブラリ、フレームワークを提供
 - 早期成果創出のためにキラーアプリケーションの同時開発を通してシステム設計に反映

	汎用コア (Server)	演算加速コア (GPU, SIMD)
データ構造	不規則・非構造	規則構造
性能電力比	低 Ex. 4.27GF/W, 22nm	高 Ex. 7GF/W, 28nm
メモリ階層の 現状	キャッシュと主メモリ	GPU側とホスト側メモリ

将来マシンイメージ群

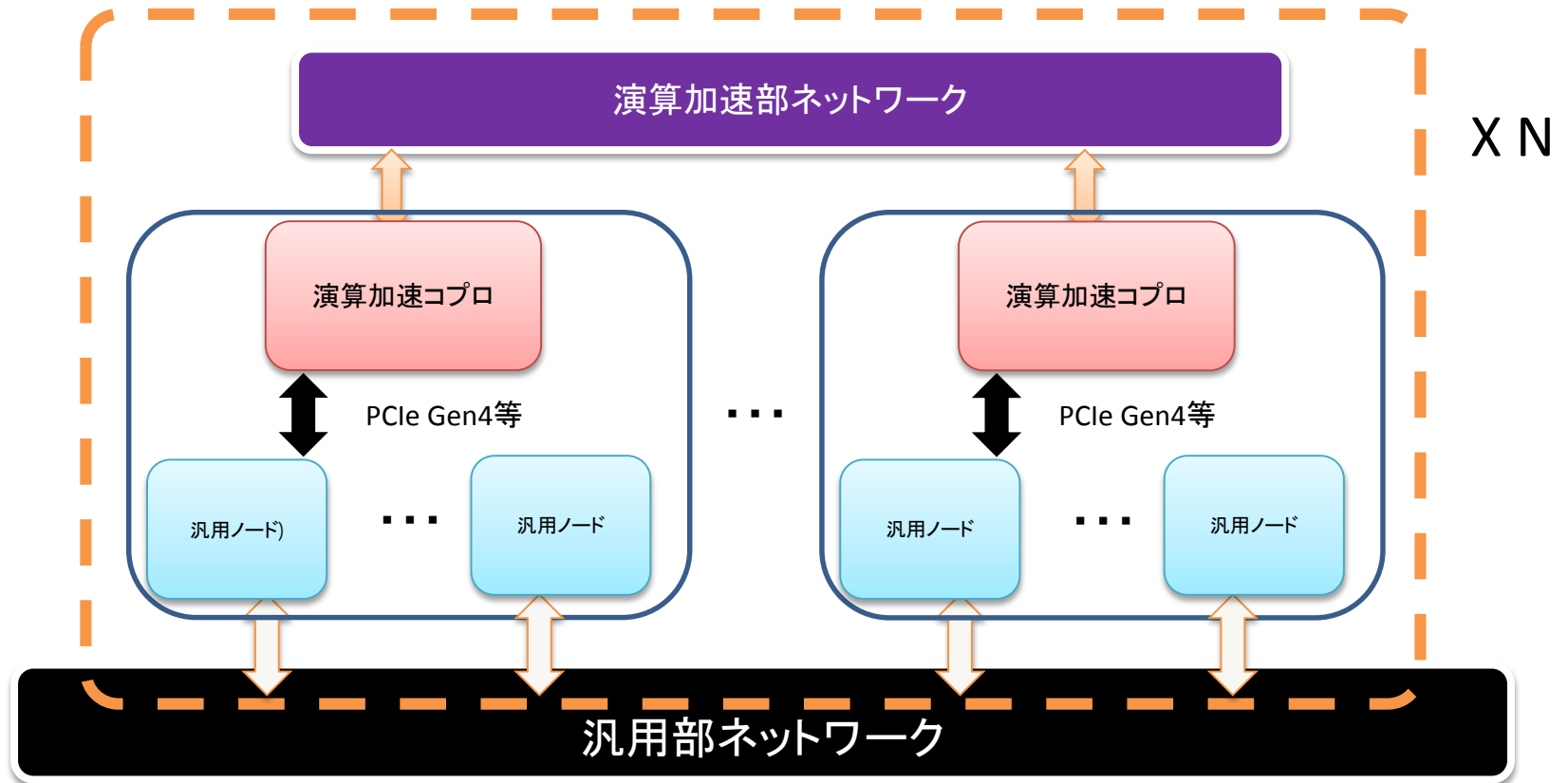
- 汎用コア、演算加速コア、ネットワークが統合されていく
- 統合方法および搭載メモリ容量とメモリバンド幅は、実装技術によって変わっていく
- 3D実装可能メモリ容量以上が必要な場合、低速メモリが導入される



MCMで演算加速コアと汎用コアを統合 演算加速コア中心 同一プロセッサの中での統合

汎用部・演算加速部構成

- 汎用部
 - 京で開発されたアプリケーションの継承を重視したCPU
- 演算補助部
 - SIMDアーキテクチャによる電力性能に優れたCPU



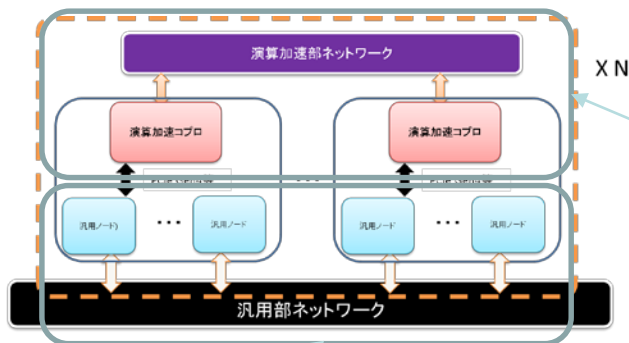
システム利用イメージ

Offloading Model

- OpenACCコンパイラディレクティブ等用いてコードの一部を演算加速コ・プロセッサにオフロード
- 演算加速コ・プロセッサで動作するライブラリを呼び出す

適応可能アプリケーション

- 量子化学など対角化や直交化が必要なアプリ
- 原子核のシェルモンテカルロ等
- UT-Heartなど大規模な行列を多数反転するアプリ
- 大規模粒子系

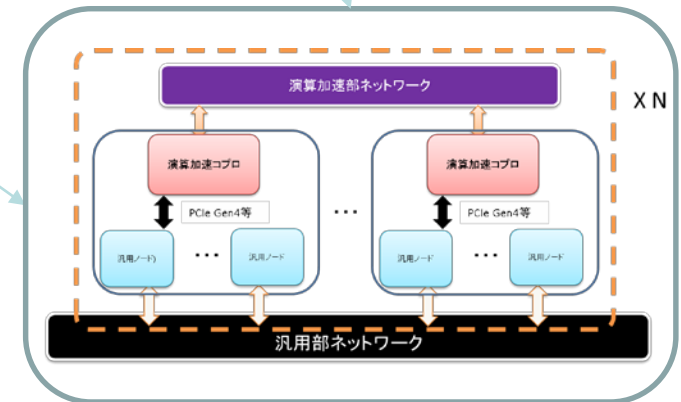


Cooperation Model

- 汎用部と演算加速部それぞれで協調するアプリ実行
- 複数の汎用部プログラムと複数の演算加速部コプロプログラムによる協調動作

適応可能アプリケーション

- 前処理を汎用で、カーネルを加速機構で実行
- 連成計算等のMPMD実行(例:気候、気象のカプラー)



Separation Model

1. 汎用部だけで計算するアプリ
2. 演算加速部のグループで実行されるアプリ
適応可能アプリケーション
 - 主に、ステンシル計算(メモリに入る場合)、QCD等
 - 分子動力学計算(MD)

設計開発基本方針

課題：2020年から運用可能な高い性能電力比と幅広いアプリケーション実行環境を有するエクサスケールマシンの実現

計算機アーキテクチャ基本方針

汎用コアと演算加速コアを有するマシン

- ✓ 汎用コアでないと性能がだせないアプリ
 - ✓ 演算加速コアで性能だせるアプリ
 - ✓ 両コアを使って性能がだせるアプリ
- を棲み分け、全電力時間積削減

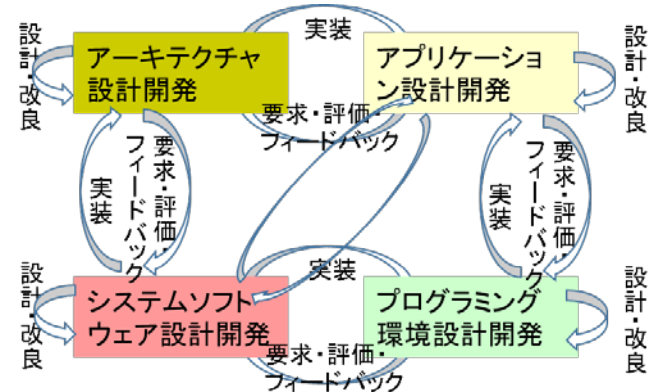
Co-design (協調設計)

- ✓ アプリケーションプログラムと計算機アーキテクチャの協調設計：演算性能・並列性能
- ✓ アプリケーションプログラムとプログラミング環境の協調設計：記述性・並列性能
- ✓ アプリケーションプログラムとシステムソフトウェアの協調設計：ファイルI/O性能・通信性能
- ✓ プログラミング環境とシステムソフトウェアの協調設計：ファイルI/O機構API・通信機構API
- ✓ 計算機アーキテクチャとシステムソフトウェアの協調設計：ファイルI/O性能・通信性能

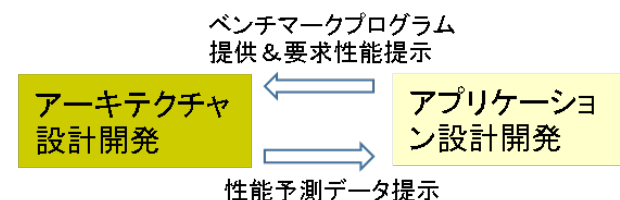
プログラミング環境設計基本方針

- ✓ 将来にわたって有効な統一プログラミングモデル、ライブラリ、フレームワークを提供
- ✓ 国際連携によるソフトウェア資産の国際的共有化促進

Co-design

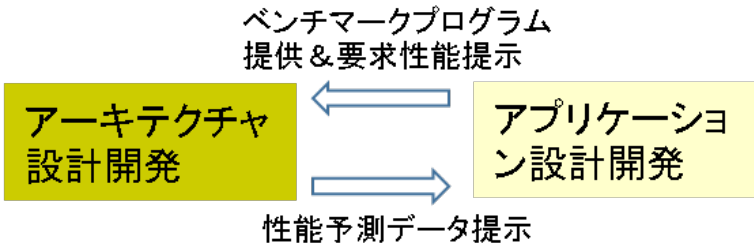


従来手法

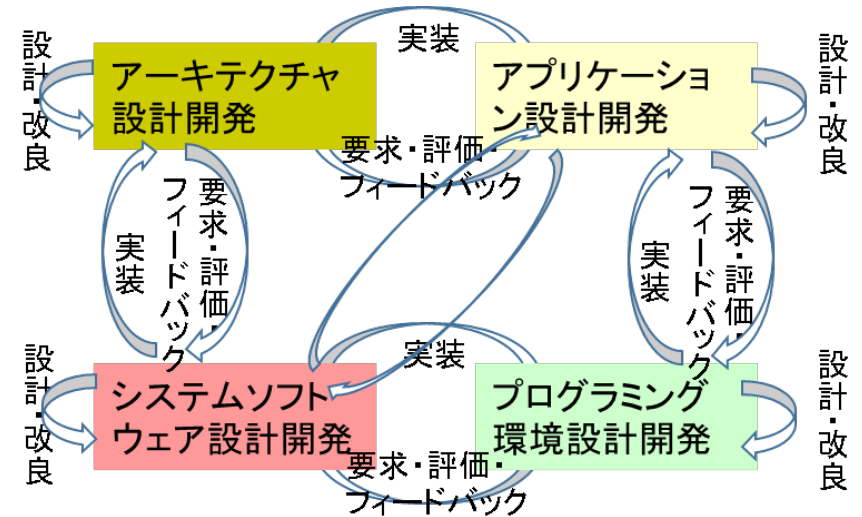


Co-Design

京（従来手法）



Co-design



「京co-designではベンチマークプログラムに対するマシンの最適化を行った。マシンができてからアプリケーションの高速化のためにアプリケーションの最適化を行っている。

システム全般に渡ってアプリ設計と相互にフィードバックしながら設計開発を同時に進める。

Co-design例：

アプリケーションのファイルI/O処理部分プログラムを見直すとともに、ストレージ系ネットワークバンド幅、ファイルシステム性能仕様を決め、システムソフトウェアのファイルI/O処理機構を設計実装評価改良をアプリ&システム側で同時に行う

目標性能：京コンピュータの100倍性能を目指す

アプリケーション実効性能で100倍

	京	ポスト京
性能	10.51 Pflops (Linpack)	エクサスケール
消費電力	12.7 MW(Linpack)	30 ~40MW

開発目標達成検証法（設置時）
社会的・科学的課題を解決するために使用されるアプリケーションの中からベンチマークプログラムとして選定し目標性能を検証する

100倍向上するアプリケーション例

医療・創薬	新薬スクリーニングのための分子動力学シミュレーション
総合防災	被害予測のための地震波伝播シミュレーション
基礎科学	量子色力学から原子核・宇宙初期の解明を目指す格子QCD計算

注1：現在概念設計レベルのため、性能および消費電力は今後精査

注2：評価に用いるベンチマーク群はアプリF Sの意見を反映しながら選定する

注3：平成26年7月までに選定ベンチマークプログラム群を用いて概念設計レベルでの性能推定を実施

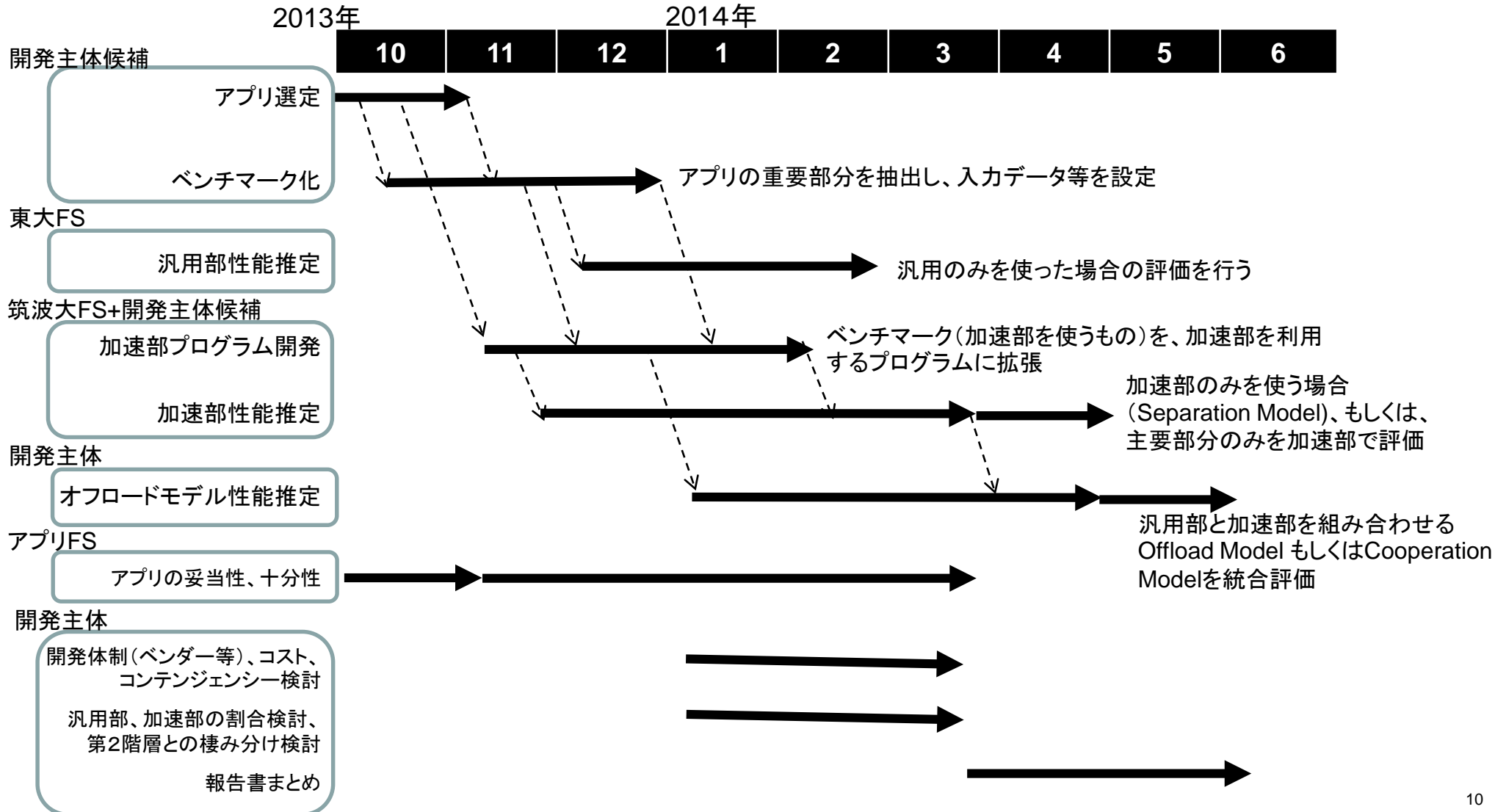
マイナビニュース

【コラム】スパコン最大の学会「SC13」に見る先端技術

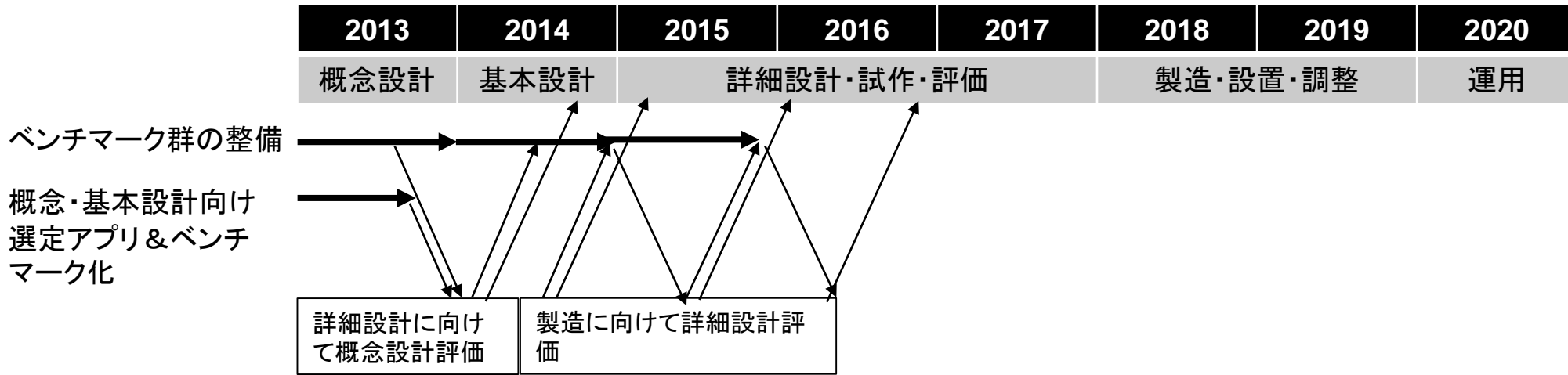
<http://news.mynavi.jp/column/sc13/007/index.html>

ポスト京概念設計性能推定等工程表

方針：社会的課題を含む課題を、計算科学ロードマップ白書(第2版)から選定。ベンチマーク化し、汎用部および加速部を用いた評価を進める。ファイルI/Oに関しては2014年度実施。



課題解決アプリ・ミニアプリ・概念設計向け選定アプリの関係



ベンチマーク群の選定方法

現行のミニアプリ群とフルアプリ群(ミニアプリ化できてないもの)からベンチマーク群を作る。

