

2012年 EV・HEVワークショップ

HILS等リアルタイムシミュレーションにおけるプラントモデルの 高速化手法

ーリアルタイムLinuxベンダーからのご提案

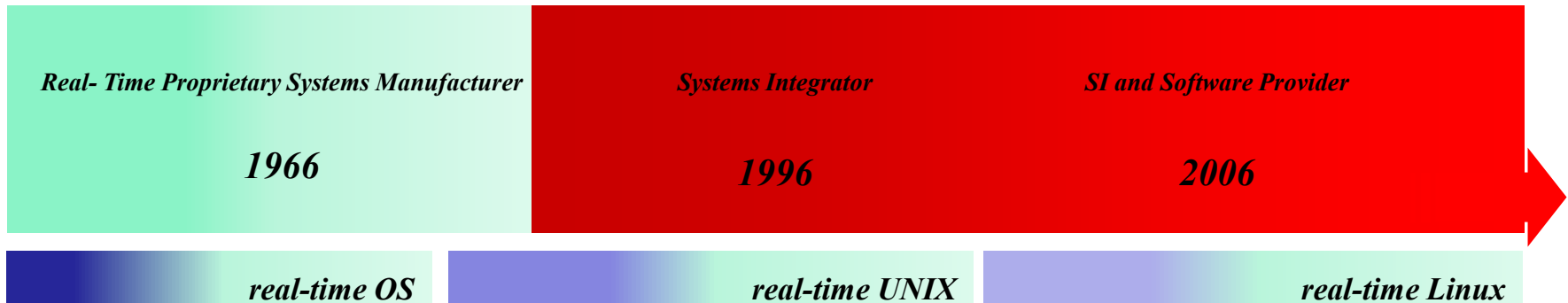
コンカレント社の提供する
リアルタイムシミュレーション・ソリューション

- 会社説明
- リアルタイムとは
- プログラム(モデル)の高速演算へのアプローチ
 - ◆ ハードウェア的アプローチ
 - ◆ 最適化(Optimize)の効果
 - ◆ リソース・アクセスプロセス
- 展示事例紹介

■ 会社説明

■ コンカレント社 会社概要

◆ 約半世紀に及ぶリアルタイム・ソリューションの取り組み



◆ リアルタイム事業部 取り扱い製品

- リアルタイムOS : RedHawk Real-time Linux OS
- リアルタイム デバッグ/分析ツール : NightStar Tools
- HILSプラットフォーム : SIMulation Workbench
- RedHawk Linux搭載リアルタイムコンピュータ : iHawk シリーズ
- RedHawk Linux搭載 イメージ出力コンピュータ : ImaGenシリーズ
- 各種ドライバ作成・動作確認などのコンサルタント
- ソフトウェア長期サポート

■ RedHawk Real-time Linux OS(2002年販売開始)



- ◆ Red Hat Enterprise Linuxと互換性を持つリアルタイムLinux OS
- ◆ 10 年以上に及び最新のLinuxカーネルに追従
- ◆ ワールドワイドで数千に渡るお客様プロジェクトでご採用
- ◆ 時間制約の厳しい(ハード・リアルタイム)分野へLinuxの汎用性を供給
- ◆ オープンソース OS(GPL)
- ◆ 市販品のハードウェア使用の経済性
- ◆ 航空宇宙・防衛分野の現場で実証された実力
- ◆ ハードウェア条件が整えば、15マイクロ秒の応答を保証

■ RedHawk Linux搭載リアルタイムコンピュータ : iHawk



- ◆ COTS(汎用市販品) 使用の経済性と良好な調達性
- ◆ 専用Hardware不要
- ◆ CPUや筐体など、数百の組み合わせから選択できる柔軟性
- ◆ 選択時点で最先端のハードウェアを使用可能

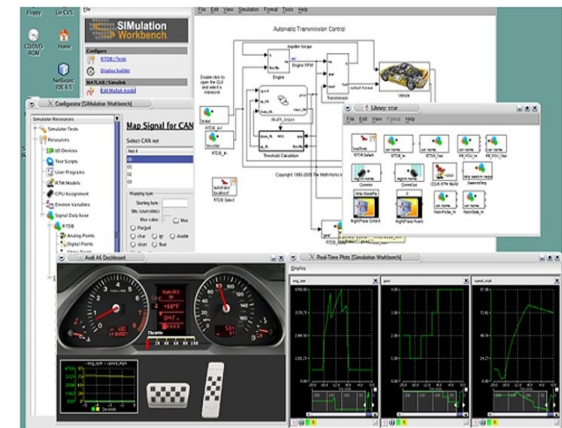


コンカレント・コンピュータは、DELL コンピュータ社・SuperMicro社等からハードウェアのOEM供給を受け、RedHawk Real-time Linuxをバンドルしたコンピュータを『iHawk』ブランドで販売しております。

■ リアルタイムHILSプラットフォーム : SIMulation Workbench



- ◆ MATLAB/Simulink及びC言語表記されたモデルを、自由に接続・交換・追加が可能
- ◆ ハードウェア制御のプログラムは一切必要ない
- ◆ コントローラは、iHawk(性能とコストに見合った市販のハードウェア)
- ◆ リアルタイムのデータ収集とそのデータ表示
- ◆ 収集後のデータ表示と、データ形式変換
- ◆ マルチCPUおよびマルチコア(Core)へ自由に実行定義



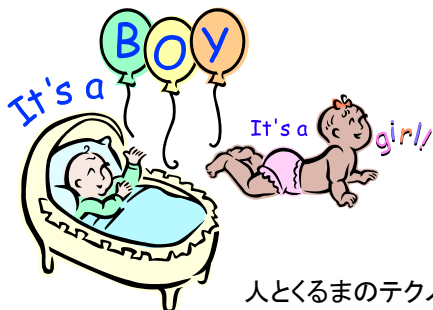
■ サポート

◆ OS・ドライバ共、自社開発による良好なサポート

- 最新のLinuxカーネルに追従(6ヶ月遅れポリシー)
- 新規I/Oボードのドライバ開発
- システム開発の請負

◆ ソフトウェアの長期サポート・ポリシー

- OSの一般サポート停止後も引き続きサポート
- 操縦訓練用シミュレータ(防衛庁様納入)など、20年以上継続中
- ハードウェア保用品調達・管理の長期契約など柔軟にサポート



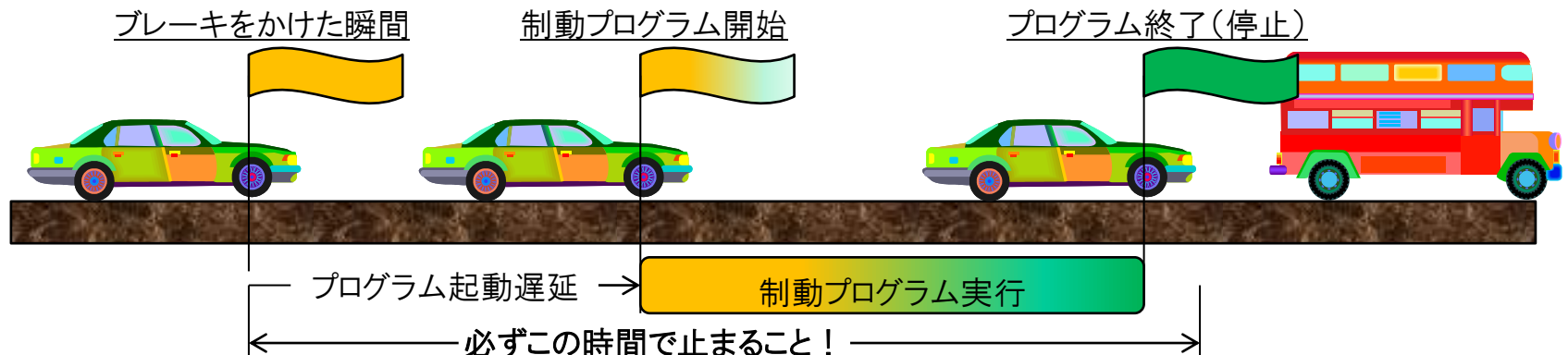
■ リアルタイムとは

■ リアルタイムの定義

◆ 日本工業規格(JIS)によれば(JIS X 0010, 日本規格協会, 1987年), 「**計算機外部の別の処理と関係を持ちながら, かつ外部の処理によって定められる時間要件に従って計算機が行うデータの処理に関する用語**」と定義

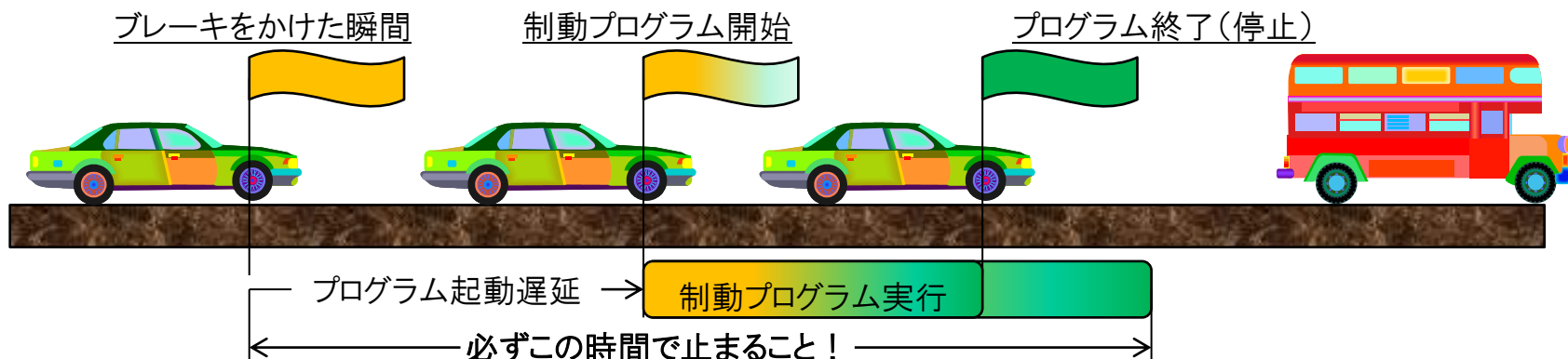
◆ 簡単な例、

- 計算機外部の別の処理と関係 : 運転者の操縦など計算機以外の状況・動作
- 外部の処理 : 急制動(急ブレーキを踏む)
- 定められる時間要件 : 決められた時間で制動プログラムが終了

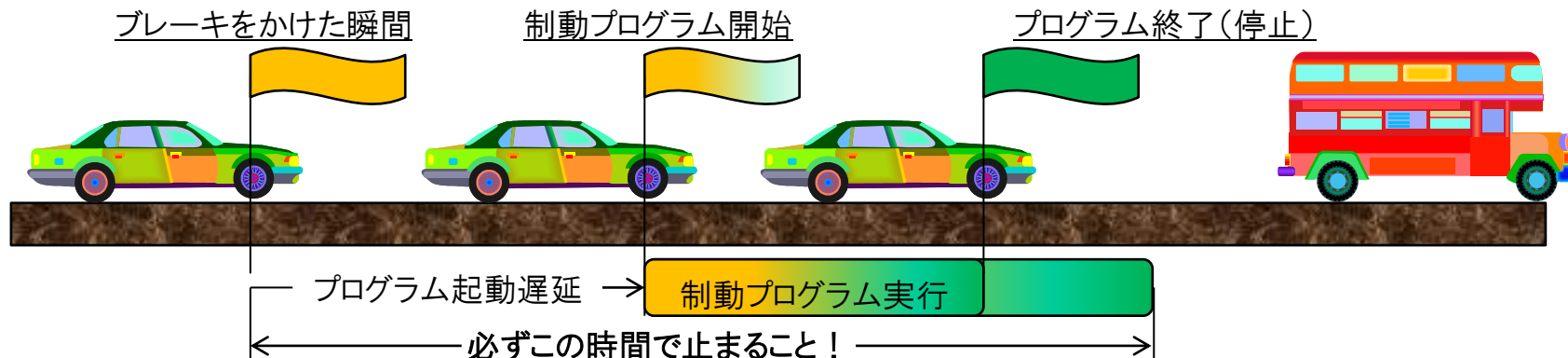


■ どの様なシステムが好ましいのか？

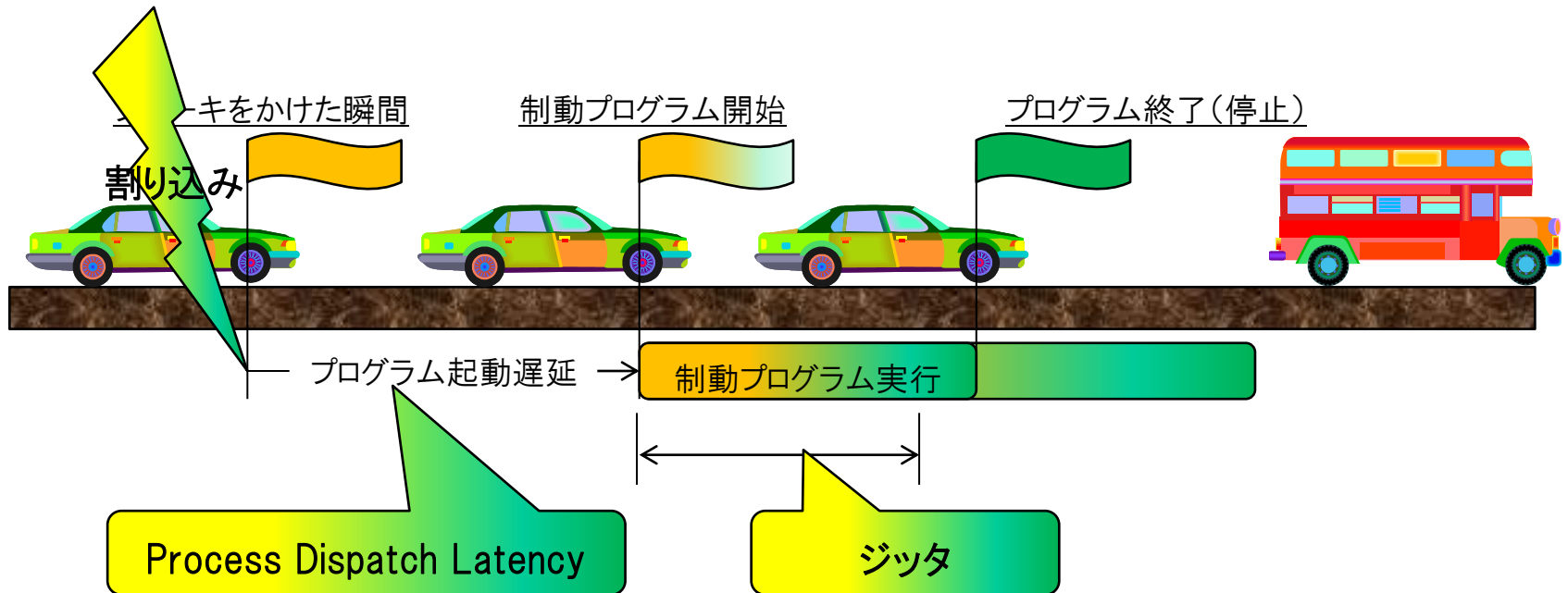
◆ 処理速度が速いシステム・・・だけ



◆ 処理速度が速いシステム・・・かつ、遅延のジッタが一定で小さい



リアルタイムOSの場合(シミュレーション=周期実行)



- ◆ OSに対してプログラムを起動するきっかけ → 割り込み
- ◆ 割り込み受信からプログラムが起動するまでの遅れ時間
→ Process Dispatch Latency
- ◆ Process Dispatch Latencyのばらつき → ジッタ

■ 良いリアルタイムOSとは・・・

◆ Determinism(決定論)

- リアルタイムシステムにおいて最も重要な性能
- 同じ処理であれば常に同じ処理時間を保証する性能のこと
- 特にシミュレーションなど周期実行システムでは重要
 - ✓ 『通常は10マイクロ秒の応答時間だが、一か月に一度程度500マイクロ秒になる』では、良いDeterminismとは言えない

◆ ハードリアルタイム・システム

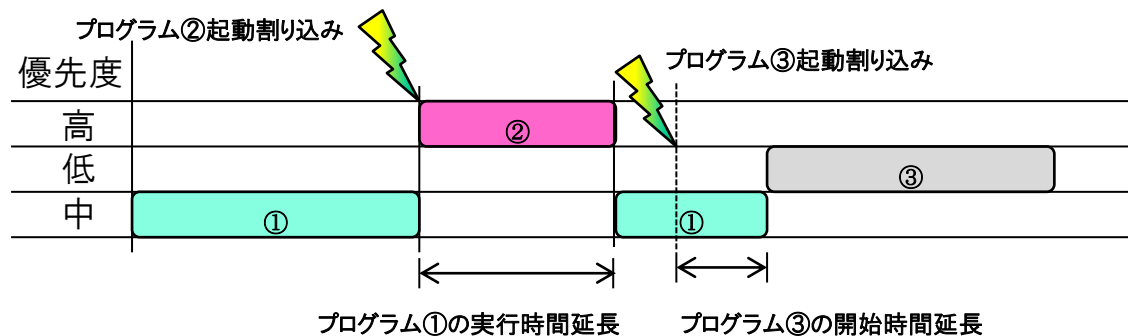
- 航空宇宙・防衛・自動車など、周期時間内に処理が完了しないと取り返しのつかない損害が発生するシステム
- 時間を守ることが最優先事項
 - ✓ 応答時間の長短がハード or ソフトを決定しない
 - ✓ 処理時間の応答時間が保証できるかが論点

■ Determinism を阻害する要因

◆ プログラムの待機時間の影響

➤ マルチタスクのための多重割り込み許可と優先度

- ✓ 一般的なLinux:PREEMPTパッチを実装(ある程度リアルタイム性を確保)



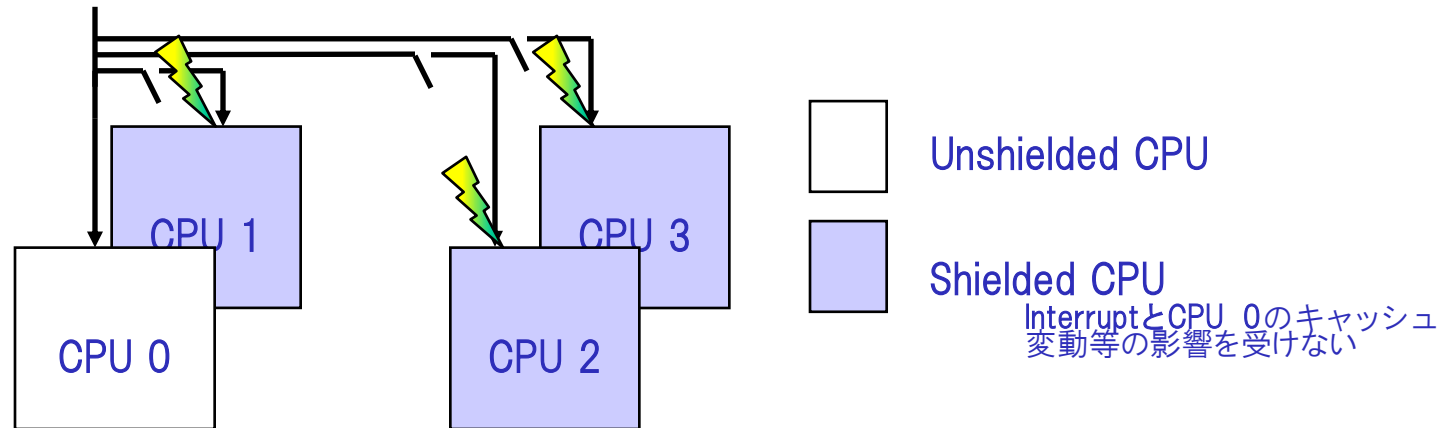
◆ Process Dispatch Latencyのジッタ

➤ 多重割り込み時の再スケジューリング

- ✓ 一般的なリアルタイムLinux → PREEMPT_RTパッチを実装
- ✓ スケジューラの起動時に多重割り込みが発生する可能性

■ RedHawk Real-time Linuxのアプローチ

緊急性の低い割り込み (キーボード/マウス、LAN etc)



◆ CPUシールド

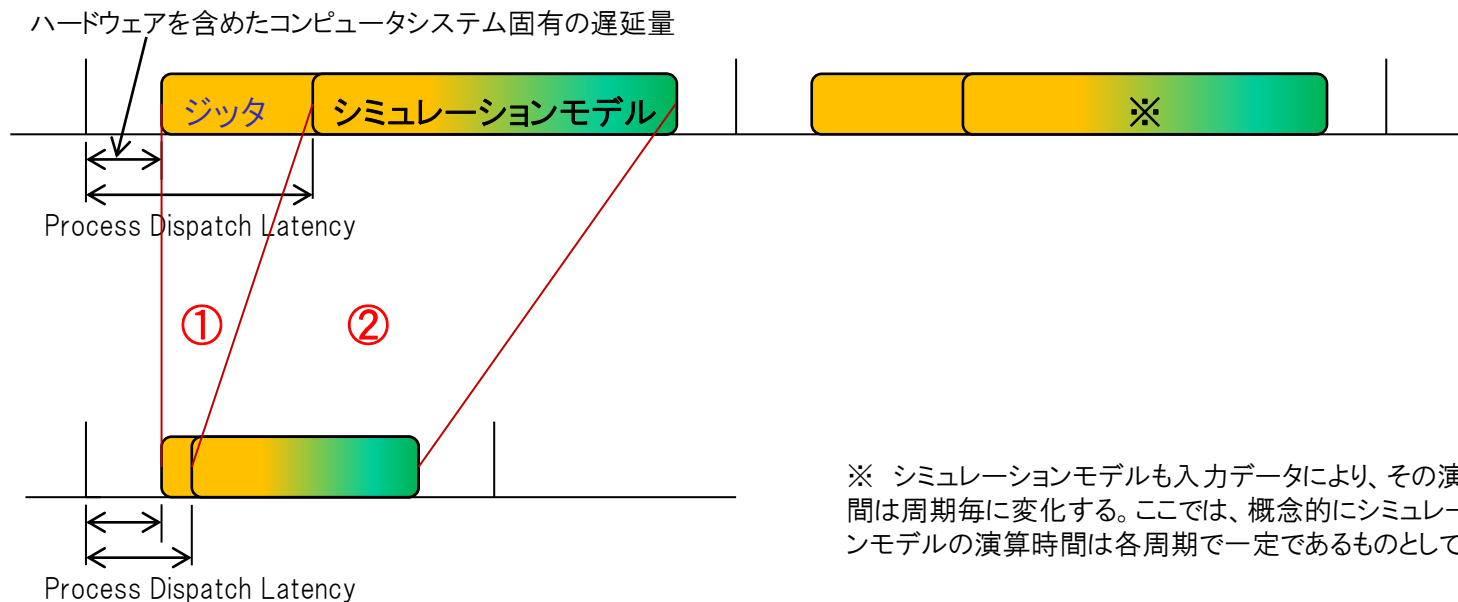
- CPUに対する割り込みを細かく禁止 → 割り込みの副作用を除去
- 複数のCPUあるいはマルチコアCPUが必要
 - ✓ 一つのプログラムを特定のCPUあるいはコアへ割り割り当てることにより、Determinismを劇的に改善 (ハードリアルタイムOS)
 - ✓ 割り込みの多重化の影響も排除可能

■ プログラム(モデル)の高速演算へのアプローチ

■ シミュレーションモデル実行の高速化

◆ 二つの方向性

- ① Process Dispatch Latencyのジッタの影響を排除
 - ✓ ジッタの生成メカニズムはOS固有 → RedHawk Linux を使用
- ② シミュレーションモデル自体の高速演算・ジッタレス

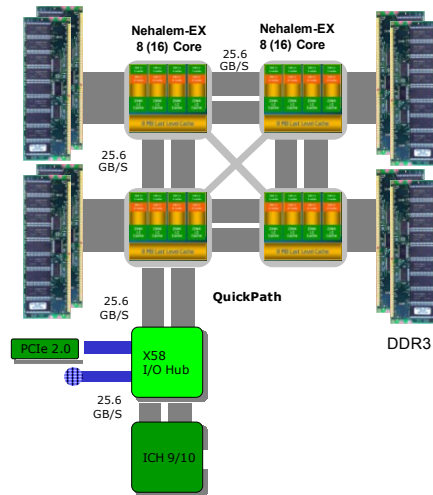


※ シミュレーションモデルも入力データにより、その演算時間は周期毎に変化する。ここでは、概念的にシミュレーションモデルの演算時間は各周期で一定であるものとしている。

- プログラム(モデル)の高速演算へのアプローチ
 - ◆ ハードウェア的アプローチ

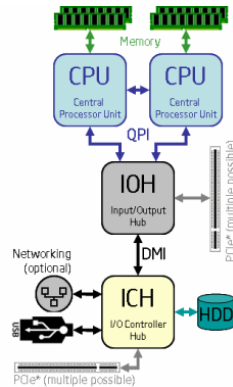
■ HILSコントローラで使用されるCPU

- ◆ メーカーの専用機が一般的
 - ユーザはCPUに関しては非常に狭い選択肢
- ◆ CPUメーカーからは順次新しいアーキテクチャの製品が発表
 - Intel 系CPUでは、1年で2倍の性能



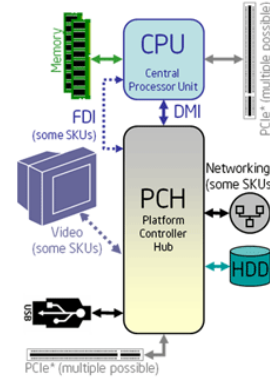
Intel QuickPath EX
4-way Xeon Beckton 4/6/8 Core

- Point-to-point high-speed links
- iHawk for next-generation Intel® 45nm microprocessor architecture (Nehalem)
- Up to 6.4 Gigatransfers/second links



5520 Tylersburg chipset

- Northbridge and Southbridge controllers
- Used on current 2-way Nehalem iHawks



3420 Ixex Peak chipset

- Single PCH controller reduces bottlenecks
- Used on current 1-way Core i5 iHawk
- Available on 2-way Industrial iHawks using Jasper Forest CPUs

■ RedHawk Real-time Linuxはオープンアーキテクチャ

◆ COTS(汎用市販品) 使用の経済性と良好な調達性

- 良好な経済性と調達性
- コンカレントは特別なハードウェアを必要としません※

✓ MathWorksNews&Notes 2011-2012 The Magazine for the MATLABR and Simulink Community__

※ コンカレント社製Real-time Clock Interrupt Module (RCIM)を使用します。

◆ 特に計算能力は、大きくCPUに依存

- 選択時点で最先端のハードウェアを使用可能
- 構成も自由



■ CPUが速くなれば、本当にモデル実行も速くなるのか？

■ コンピュータの性能比較(ベンチマーク)

◆ 米国のSPEC(Standard Performance Evaluation Corporation)の供給するベンチマーク・プログラムを使用

➤ SPECは米国カリフォルニア州に登録されている非営利団体

➤ SPEC CPU2006は、コンピュータ間のパフォーマンスを比較評価するためのベンチマーク用プログラム

✓ CINT 2006は整数演算測定用

✓ CFP2006は浮動小数点演算測定用

➤ ベンチマーク結果はweb公表
(<http://www.spec.org> 2012/05/09現在)

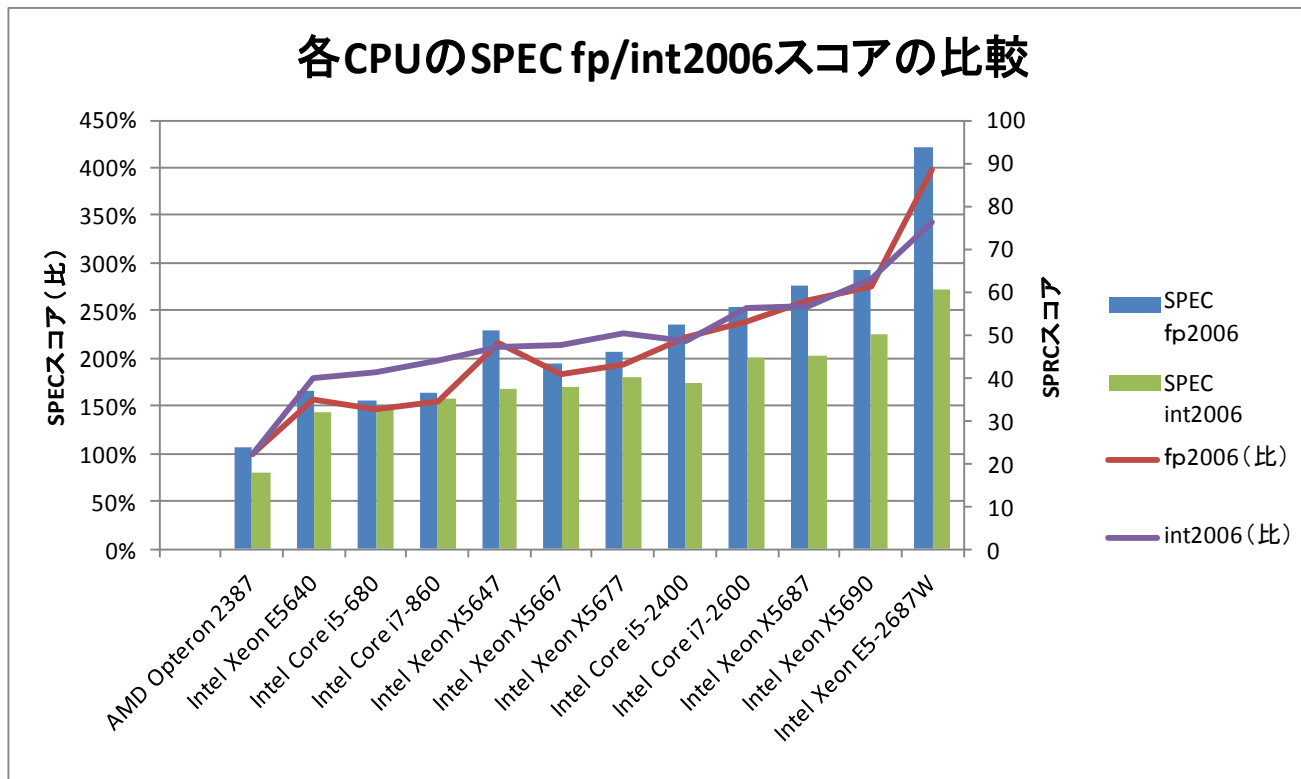


■ HILSコントローラに使用されるCPUの性能比

CPU	発売	Freq [GHz]	Cores	CFP2006		CINT2006		Memo
AMD Opteron 2387	2009/Q3	2.8	4	23.6	100%	17.7	100%	国内A社・海外D社採用のCPU同等
Intel Core i5-680	2010/Q2	3.6	2	34.8	147%	32.7	185%	海外S社最新機種採用のCPU
Intel Core i7-860	2009/Q3	2.8	4	36.3	154%	35.1	198%	海外D社最新機種採用のCPU
Intel Xeon E5640	2010/Q1	2.67	4	36.8	156%	31.9	180%	弊社で実績のあるCPU 他社製品使用CPUよりも高速
Intel Xeon X5667	2010/Q1	3.07	4	43.4	184%	37.8	214%	
Intel Xeon X5677	2010/Q1	3.46	4	45.8	194%	40.2	227%	
Intel Core i5-2400	2010/Q1	3.1	2	52.4	222%	38.5	218%	
Intel Xeon X5647	2011/Q1	2.93	4	50.9	216%	37.5	212%	弊社で実績のある、比較的新しいCPU 他社製品使用CPUよりも高速
Intel Core i7-2600	2011/Q1	3.5	4	56.5	239%	44.6	252%	
Intel Xeon X5687	2011/Q1	3.6	4	61.6	261%	45.1	255%	
Intel Xeon X5690	2011/Q1	3.46	6	65.1	276%	50.1	283%	
Intel Xeon E5-2687W	2012/Q1	3.1	8	93.8	397%	60.6	342%	現時点でiHawkで使用可能な最高スペックCPU

- ◆ 表のSPEC項の左側にはSPEC社より公表されている各ベンチマーク項目のスペック値
- ◆ 右側には最上段のAMD社 Opteron 2387を基準とした場合の比[%]

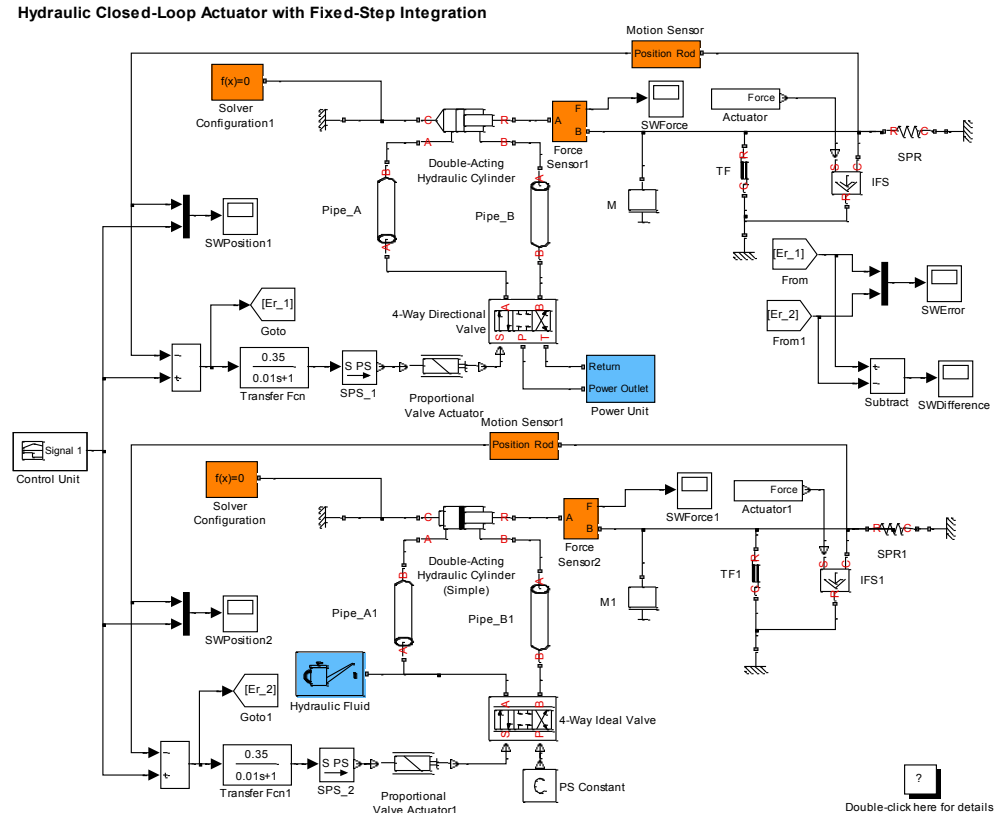
■ HILSコントローラに使用されるCPUの性能比



- ◆ 演算速度の観点からみたCPU性能は、動作周波数だけで決まるものではなくマイクロアーキテクチャに大きく依存する
- ◆ 最新のCPU搭載機では、数倍の演算能力の差が生じる場合もある

■ Simulinkモデルのベンチマーク結果：ベンチマークプログラム

- ◆ MathWorks社製
SimHydraulicsの油圧
および制御システムの
物理モデリング
- ◆ 以降モデルによるベン
チマークは、SimHydraulics
に添付されている右図の
サンプルプログラムを使用

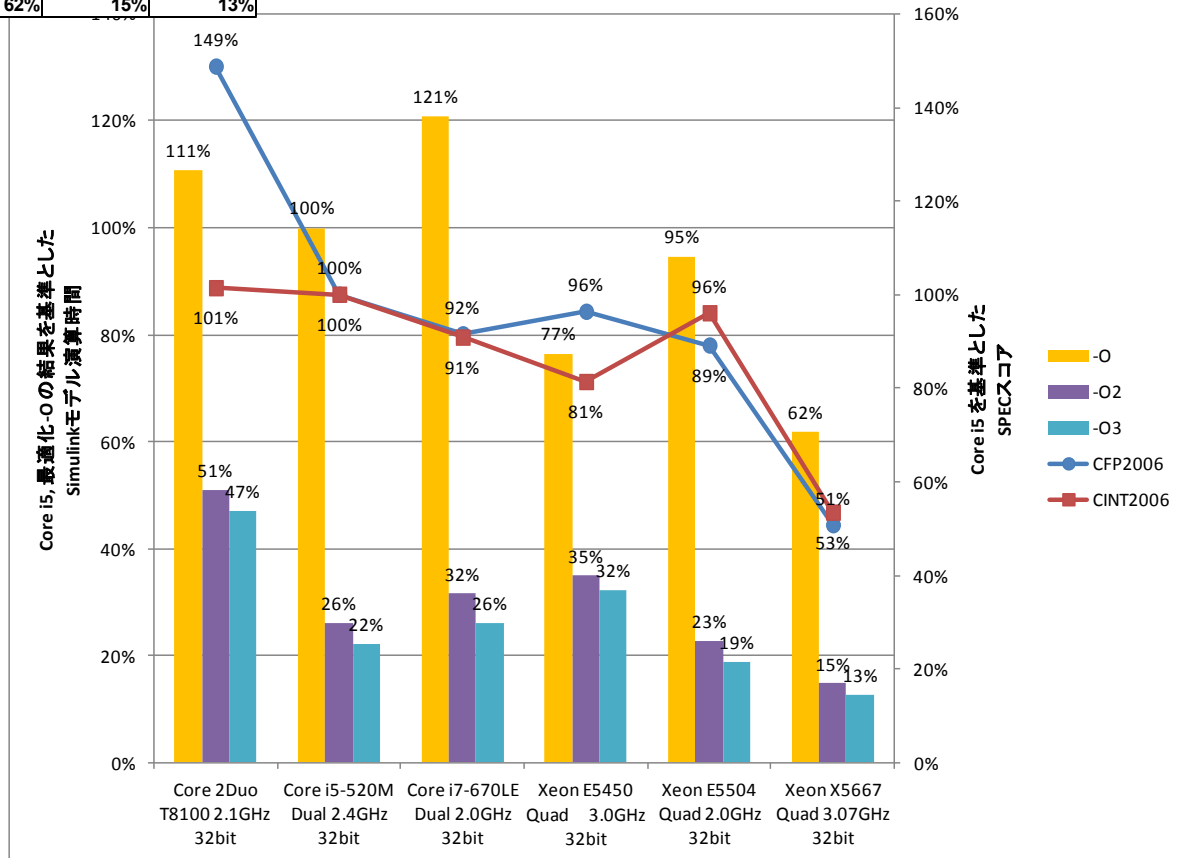


OS (Kernel) : RedHawk5.2.5 (2.6.26.8-RedHawk-5.2.5-trace, 32/64bit)
 SIMulation Workbench : Version 4.1-5(32/64bit)

■ Simulinkモデルのベンチマーク結果：CPUによる差

CPU	SPEC		gcc最適化オプション		
	CFP2006	CINT2006	-O	-O2	-O3
Core 2Duo T8100 2.1GHz 32bit	149%	101%	111%	51%	47%
Core i5-520M Dual 2.4GHz 32bit	100%	100%	100%	26%	22%
Core i7-670LE Dual 2.0GHz 32bit	92%	91%	121%	32%	26%
Xeon E5450 Quad 3.0GHz 32bit	96%	81%	77%	35%	32%
Xeon E5504 Quad 2.0GHz 32bit	89%	96%	95%	23%	19%
Xeon X5667 Quad 3.07GHz 32bit	51%	53%	62%	15%	13%

SPECスコアとSimulinkモデル演算時間の相関



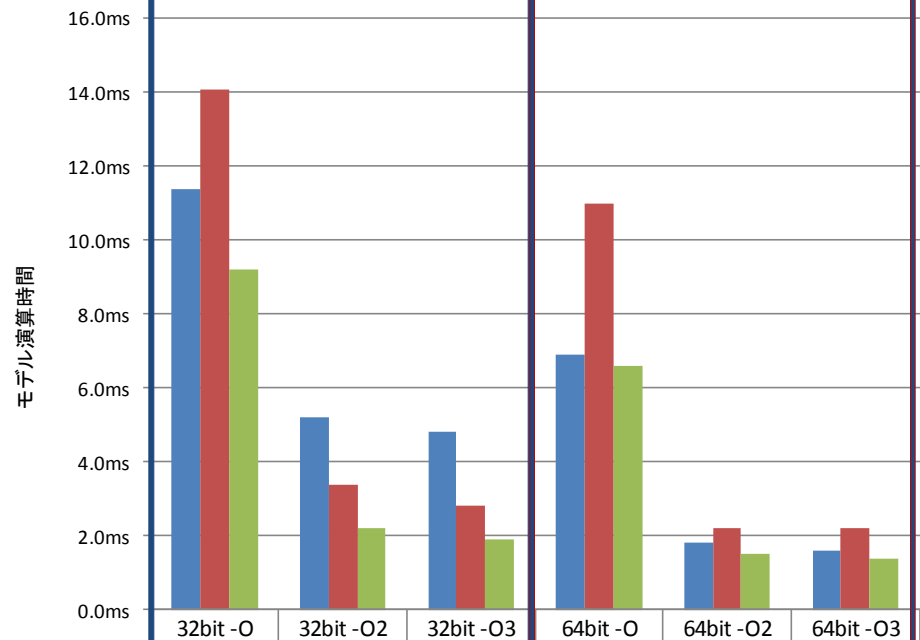
- ◆ Simulinkモデルの演算時間とSPECスコアは概ね相関する
- ◆ 最適化されたモデルが相関性が高い
- ◆ SPECスコア値の良いCPU程高速
 - SPECスコアから演算速度を類推

■ Simulinkモデルのベンチマーク結果: OS環境による差

CPU	32bit環境			64bit環境			32bitに対する64bitの演算時間の割合		
	32bit -O	32bit -O2	32bit -O3	64bit -O	64bit -O2	64bit -O3			
Xeon E5450 Quad 3.0 GHz	11.4ms	5.2ms	4.8ms	6.9ms	1.8ms	1.6ms	60.5%	34.6%	33.3%
Xeon E5504 Quad 2.0GHz	14.1ms	3.4ms	2.8ms	11.0ms	2.2ms	2.2ms	78.0%	64.7%	78.6%
Xeon X5667 Quad 3.07GHz	9.2ms	2.2ms	1.9ms	6.6ms	1.5ms	1.4ms	71.7%	68.2%	73.7%

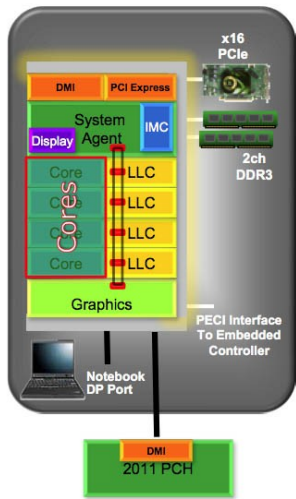
- ◆ CPUにより効果の差はあるが、20%~30%の高速化を観測
- ◆ 最適化オプションによっても差
 - gccでは-o2が比較的64bit化の効果が高い

32bit/64bit環境の違いと最適化(Optimize)の影響



Xeon E5450 Quad 3.0 GHz	11.4ms	5.2ms	4.8ms	6.9ms	1.8ms	1.6ms
Xeon E5504 Quad 2.0GHz	14.1ms	3.4ms	2.8ms	11.0ms	2.2ms	2.2ms
Xeon X5667 Quad 3.07GHz	9.2ms	2.2ms	1.9ms	6.6ms	1.5ms	1.4ms

RedHawk Real-time Linuxの対応する高性能CPU



Sandy Bridge
offers significant improvements
over Nehalem 5500/5600

model	対応ソケット数	Speed (GHz)	GPU Speed (MHz)	L3 Cache (MB)	コア/スレッド数	対応メモリ (MHz)	TDP (Watts)	Package	SPEC fp2006	SPEC int2006		
E5-2687W	2	3.1	N/A	20	8/16	1600x4	150	LGA2011	93.8	397%	60.6	342%
E5-2690	2	2.9	N/A	20	8/16	1600x4	135	LGA2011	88.4	375%	58.4	330%
E5-2689	2	2.6	N/A	20	8/16	1600x4	115	LGA2011				
E5-2680	2	2.7	N/A	20	8/16	1600x4	130	LGA2011	85.2	361%	54.6	308%
E5-2670	2	2.6	N/A	20	8/16	1600x4	115	LGA2011	84.3	357%	52.5	297%
E5-2667	2	2.9	N/A	15	6/12	1600x4	130	LGA2011	85.2	361%	52.5	297%
E5-2665	2	2.4	N/A	20	8/16	1600x4	115	LGA2011	79.8	338%	49.6	280%
E5-2660	2	2.2	N/A	20	8/16	1600x4	95	LGA2011	78.3	332%	47.6	269%
E5-2658	2	2.1	N/A	20	8/16	1600x4	95	LGA2011				
E5-2650L	2	1.8	N/A	20	8/16	1600x4	70	LGA2011	62.2	264%	37.8	214%
E5-2650	2	2	N/A	20	8/16	1600x4	95	LGA2011	72.9	309%	45.0	254%
E5-2648L	2	1.8	N/A	20	8/16	1600x4	70	LGA2011				
E5-2643	2	3.3	N/A	10	4/8	1600x4	130	LGA2011	82.4	349%	50.5	285%
E5-2640	2	2.5	N/A	15	6/12	1333x4	95	LGA2011	74.1	314%	45.4	256%
E5-2637	2	3	N/A	5	2/4	1600x4	130	LGA2011	68.1	289%	45.9	259%
E5-2630L	2	2	N/A	15	6/12	1333x4	60	LGA2011	65.3	277%	38.7	219%
E5-2630	2	2.3	N/A	15	6/12	1333x4	95	LGA2011	70.5	299%	42.6	241%
E5-2620	2	2	N/A	15	6/12	1333x4	95	LGA2011	64.5	273%	38.9	220%
E5-2609	2	2.4	N/A	10	4/4	1066x4	80	LGA2011	59.2	251%	35.8	202%
E5-2603	2	1.8	N/A	10	4/4	1066x4	80	LGA2011	47.2	200%	27.5	155%

◆ 現在最もSPECスコアの高いE5-2687W

➤ SPECスコアでAMD社 Opteron 2387の4倍近い速度で動作可能

◆ Sandy-Bridgeマイクロアーキテクチャ

1. コア数の増加: 2, 4, 8コア
2. 低消費電力
3. CPUに直結されたインテグレイティッド I/O: PCIe x16

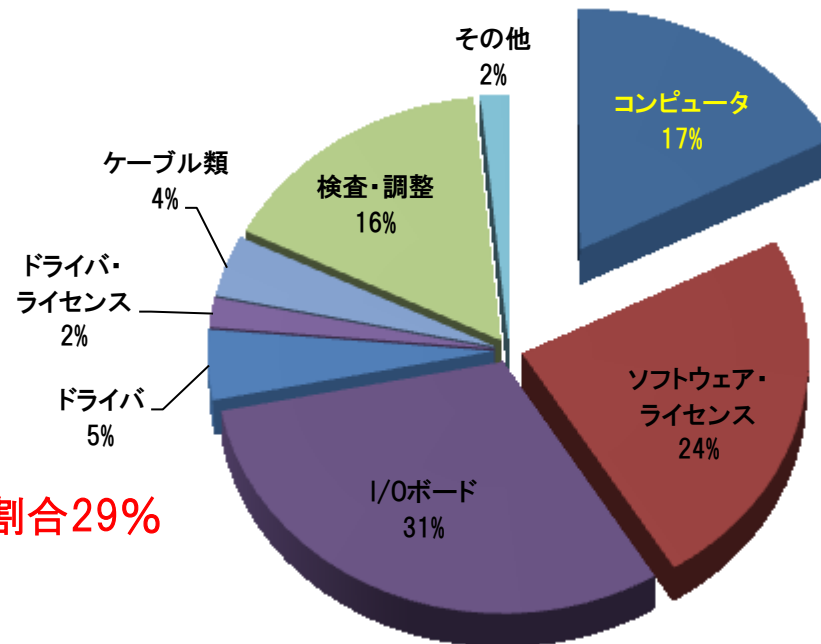
■ HILSコントローラに使用されるCPUのコスト

◆ 中規模システムのコスト全体に占めるコンピュータ(CPUは1~2個)の割合は15~20%程度

◆ 更にこれらのコンピュータのコスト全体に対して、CPUの占める割合は通常10%前後

◆ 2倍の演算性能のコンピュータを2倍のコストで換装
 ➤ 全体に占めるコンピュータの割合29%の投資で2倍の性能を入手

中規模HILSプラットフォームの
コスト内訳の一例



■ HPC(High – Performance Computing)

◆ CPUは日々進化

- マルチ・コア: Dual, Quad, 6, 10 – Core
- マイクロアーキテクチャ: Westmere, Sandy Bridge, ...

■ HILSも”HPC“環境へ

◆ iHawk 1台で可能な構成

- Intel Xeon E7-8870 10core 2.4GHz
- 最大8CPU, 80コア構成※
- メモリ最大1,024GB

※ ハイパースレッディング機能は、リアルタイム性能を損なう恐れがあるため使用していません。



■ これより上は、スーパーコンピュータ！

■ HILSコントローラに使用されるCPUの選択範囲の有効性

- ◆ ベンチマークスコアとモデル実行能力は、比例する
- ◆ ベンチマークスコアとCPUの価格は反比例する

➤ 要求性能とコストの関係上、有効な投資が可能

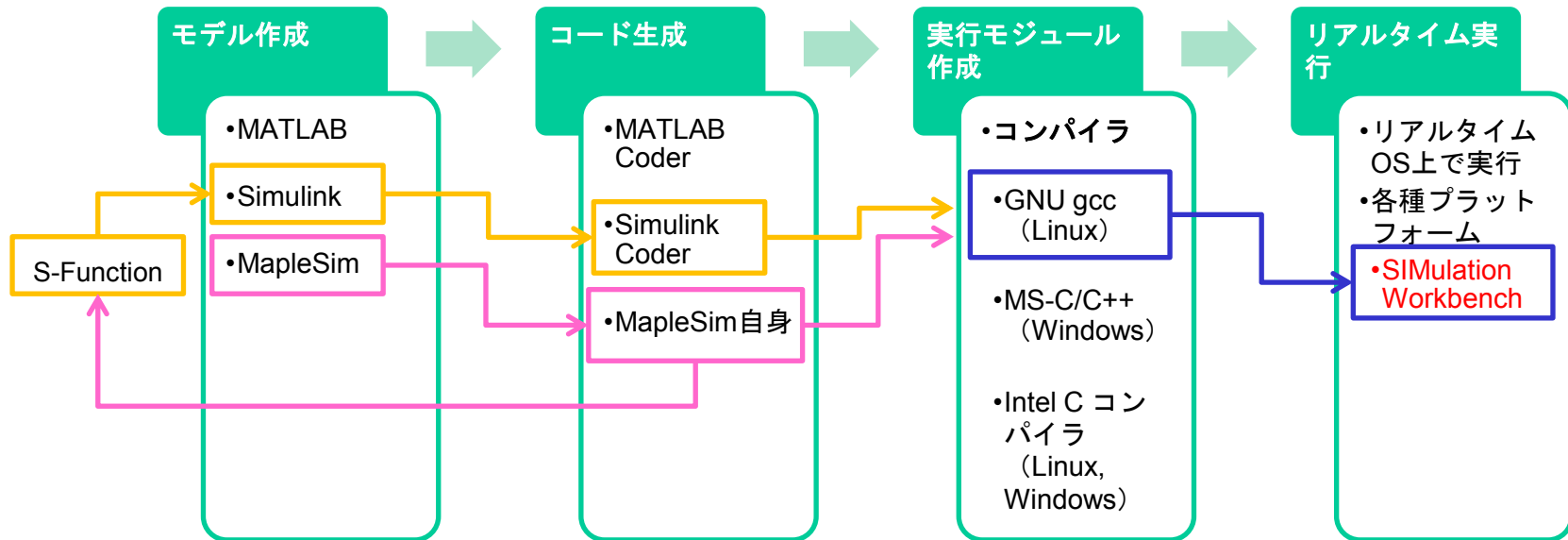
- ◆ CPUの進歩により、能力は1・2年で倍～数倍
- ◆ HILSシステムの30%前後の投資で演算性能が倍～数倍

➤ 維持費を積み重ねるよりも、換装する方が有利



- プログラム(モデル)の高速演算へのアプローチ
 - ◆ 最適化(Optimize)の効果

■ 最適化によるリアルタイム実行モジュールの高速化



◆ コンパイラの最適化(Optimize)オプション

➤ ソースコード中の冗長部分やループ最適化、レジスタ定義等

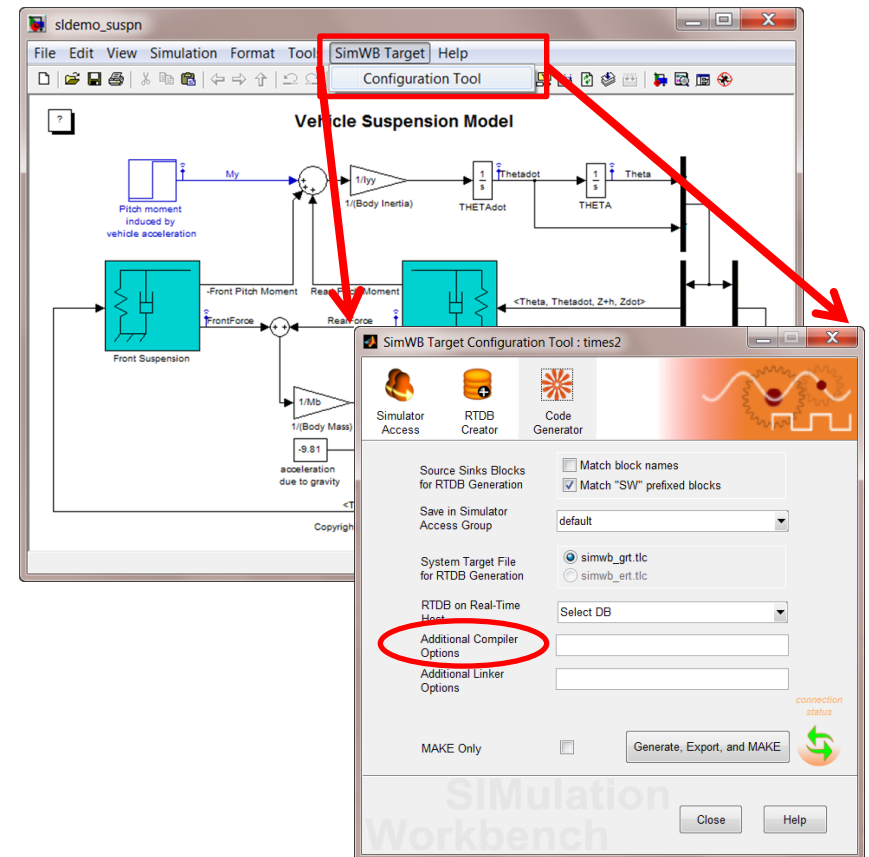
✓ 演算結果に悪影響を及ぼす場合も有 → 演算結果の精査が必要

■ フレームサイクル毎のモデル演算時間の測定

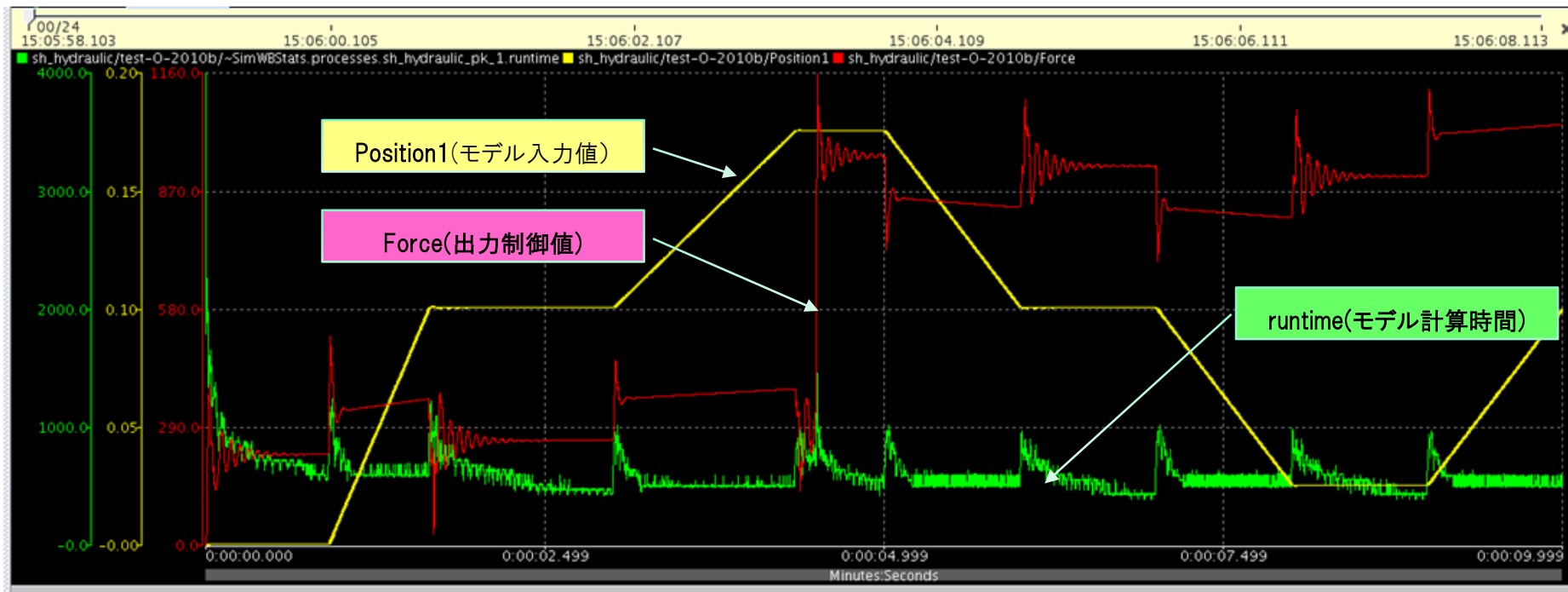
◆ コンパイルオプションの設定

- SIMulation Workbenchの Simulink Toolを使用
- 専用GUIにオプションを入力するだけ

◆ リアルタイムシミュレーションの結果は、SIMulation Workbenchのグラフ表示で確認



■ フレームサイクル毎のモデル演算時間のグラフ化



◆ 最適化オプション -0(最適化無)の場合の

- Avg.runtime: 555.09 μ sec
- Max.runtime: 14594.54 μ sec

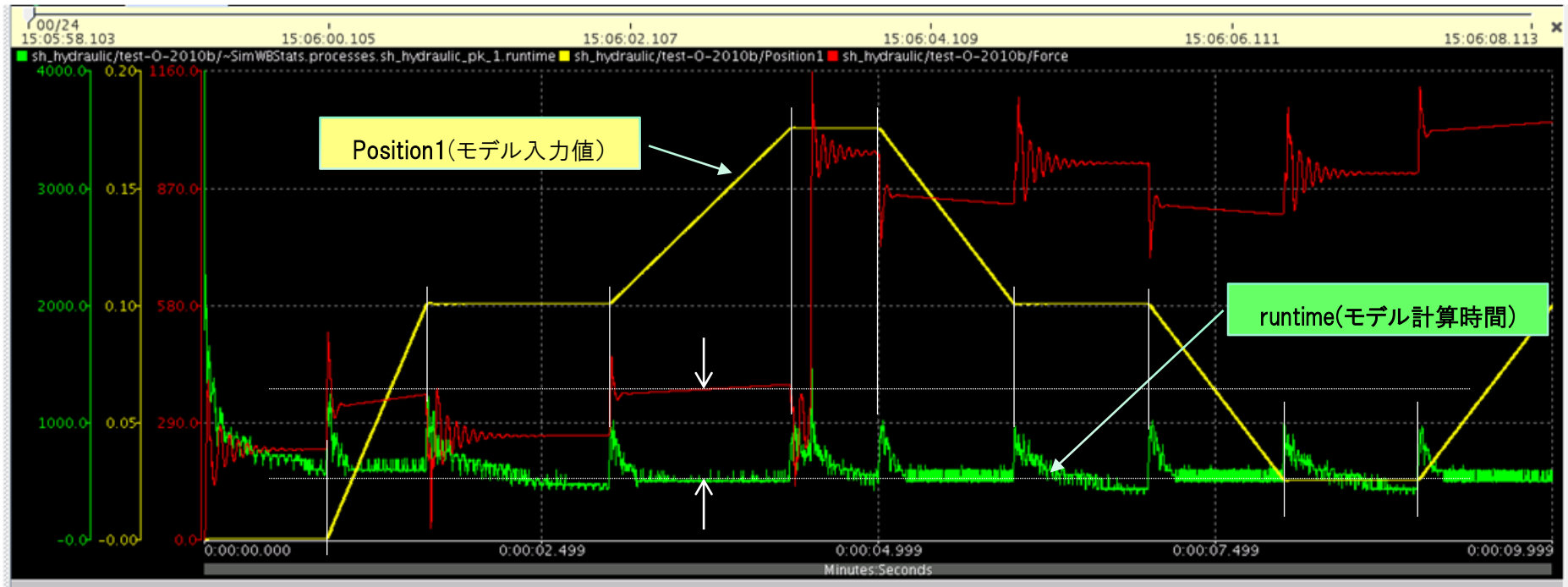
■ フレームサイクル毎のモデル演算時間のグラフ化



◆ 最適化オプション -02の場合

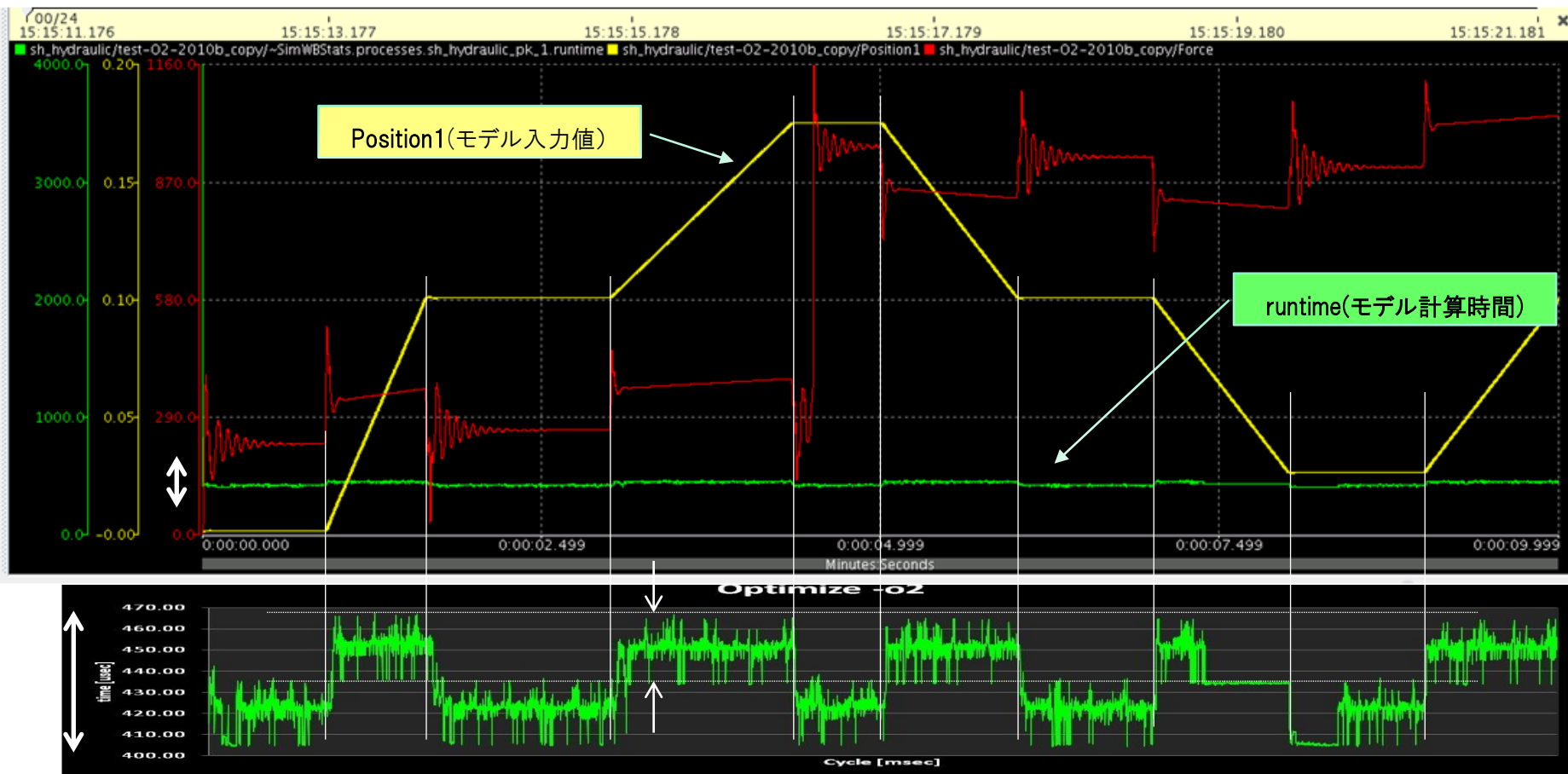
- Avg.runtime: 530.17 μ sec: **24.92 μ sec (約4.4%) 短縮**
- Max.runtime: 13766.33 μ sec

■ モデル計算時間のジッタ(最適化オプション -0)



- モデル入力値の変化点で、モデル計算時間が大きく変動
 - ✓ AverageとPeekの差 : 最大540 μ sec(合計で1msecを超える)

■ モデル計算時間のジッタ(最適化オプション -O2)



➤ モデル入力値の変化点で、モデル計算時間の若干の変動

✓ モデル計算時間のジッタ幅 : **最大56 μ sec**

■ コンパイラの最適化オプションの有効性

- ◆ 定常時のプログラム速度が高速
- ◆ 変化点の影響が極小化
- ◆ Determinism の改善に有効

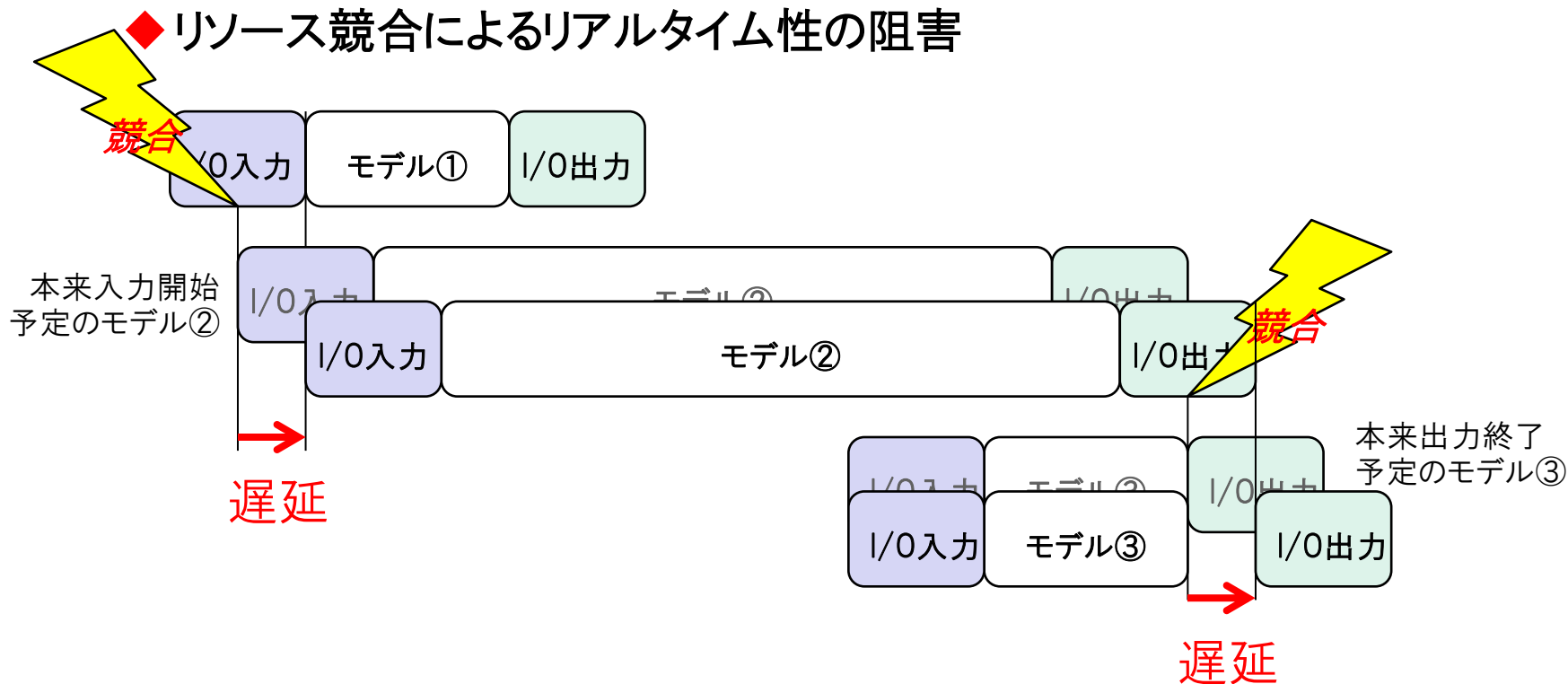
➤ モデルのアルゴリズムを変更しないで高速化

➤ 非同期に発生するオーバーラン(周期時間内に終了しない現象)の原因の一つが解決



- プログラム(モデル)の高速演算へのアプローチ
 - ◆ リソース・アクセスプロセス

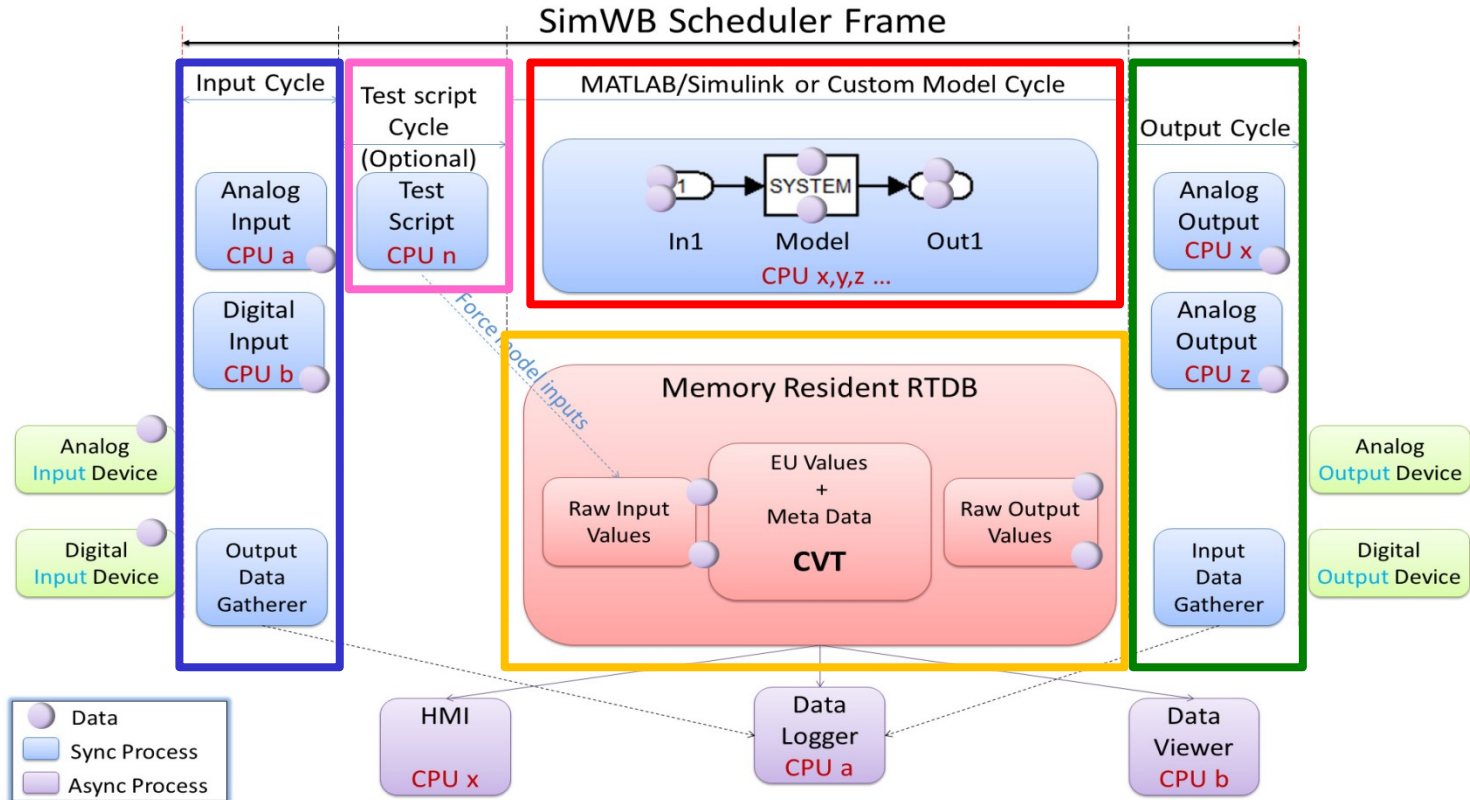
◆ リソース競合によるリアルタイム性の阻害



➤ 複数のモデルが各々同一のハードウェアI/Oを実施

- ✓ 同一リソースに対する競合が発生
- ✓ 遅延が発生する可能性が高い

■ モデルのリアルタイム実行を意識したツールを実装



◆ SIMulation Workbench のアプローチ

- リソースの入出力はシミュレーション周期の最初と最後で一括実行
- 同一リソースに対する競合・遅延発生しない

■ SIMulation Workbenchのリソース・アクセス方式の効果

- ◆ 各モデルがI/O入出力部を持つ → リソース競合・遅延の発生
- ◆ リソースの入出力はシミュレーション周期の最初と最後で一括実行

➤ 同一リソースに対する競合・遅延発生しない



- ◆ モデルとI/Oインタフェースを切り離す
- ◆ モデルの入出力を論理的なシンボルのままリアルタイム実行が可能

➤ 開発環境とリアルタイム環境で、モデルは同一プログラム
✓ 特にSimulinkでは、コンパイルまでSimulink環境実施

■ 高速かつ高性能プラットフォームの持つベネフィット

◆ 2倍の計算速度を持つCPUがあれば…

- より複雑なモデルを実行できます
- 環境に併せてモデルを制限する必要はありません

◆ 同じコストで、コア数が2倍あれば…

- もっと多くのモデルが並行して実行できます
- 安価に複数の物理領域に渡る物理システムの統合の可能性

◆ 多少能力を制限しても安価な構成が出来れば…

- 一人に一台のシミュレータも夢ではありません

◆ 既存のHILS環境や試験装置と結合出来れば…

- 既存設備の有効利用ができます
- 他に類をみないシミュレーション環境の構築ができます

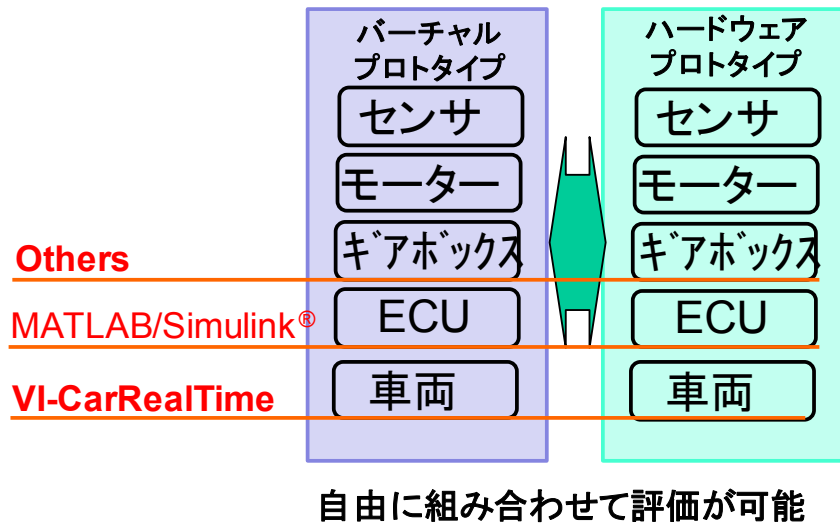
■ どの様なことでも、ご相談ください

■ 展示事例紹介

■ 事例1:ステアリングHILS

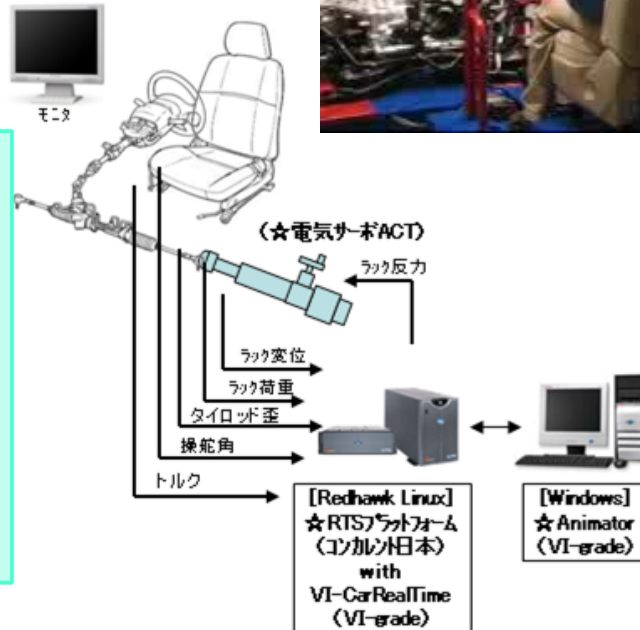
◆ パワステ開発用HILS

- ECU適合
- リアルな操舵力を再現
- 主観的評価が可能



<システム概要>

★ステアリングHILSシステム



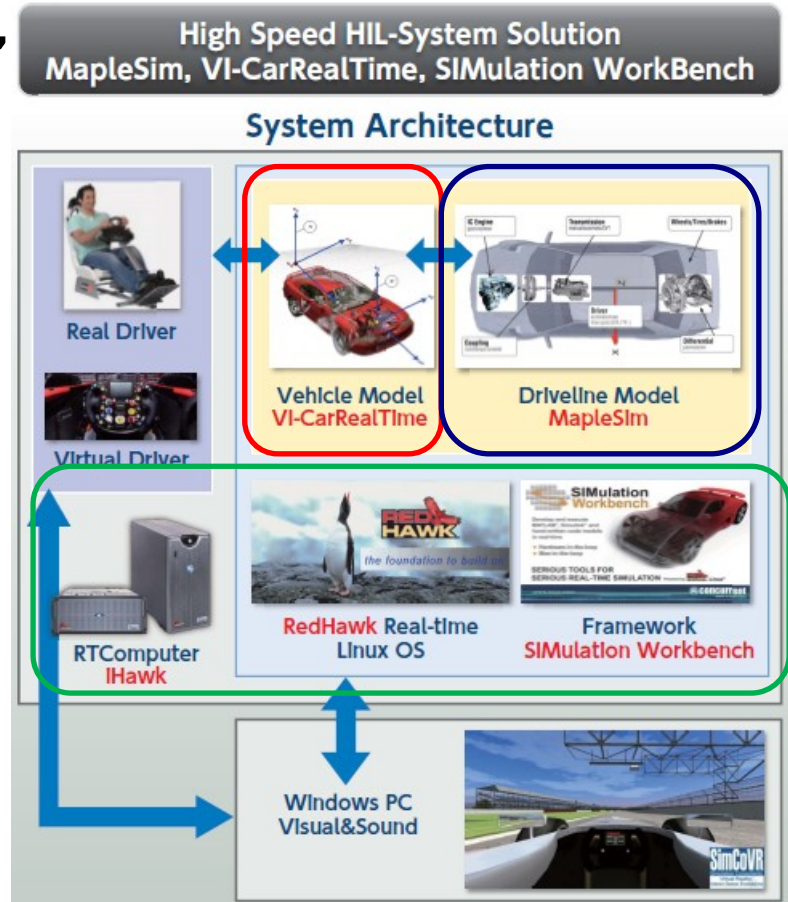
◆ 鷺宮製作所様展示ブースで

■ 事例2:EV/HEVリアルタイムシミュレーション

◆ “複数の物理モデルを組み合わせた”
高速なHILシステムソリューション

- 車両運動モデル :VI-CarRealTime
- EVモータ :MapleSim
- HILS環境 :RedHawk

◆ コンカレント日本、
サイバネットシステム様
日本ヴィアイグレード様による
3社共同展示ブースで試乗可能



ご視聴ありがとうございました。



<http://www.ccur.co.jp>

【E-mail】 info@ccur.co.jp