Linuxカーネル状態トレーサ
LKST
（Linux Kernel State Tracer）

（株）日立製作所

杉田由美子、畑崎恵介

開発の背景

● エンタープライズLinuxへのRAS強化要求
ーミッションクリティカルな処理を実行できる高信頼性ー

いつ起きるかわからない障害の検出
発生後の早期解決(ダウンタイムは最小限に)

◆運用中にも情報採取できるツールが必要
◆エンタープライズ**Linux**への**RAS**強化を4社協業(IBM・NEC・日立・富士通)にて推進

RAS:reliability, avail ability, and serviceability

RASツールに求められるもの

- システムを停止させることなく色々な情報を収集
  - 障害状況にあわせた採取情報の変更
  - 障害発生過程が判る情報の採取
- システムがクラッシュした過程情報も収集
- オーバヘッドは少なく
- 大規模SMPにも対応

⇒ **Linux カーネル状態トレーサ（LKST）の開発**

LKST の特徴

- 充分な情報の採取
  - 重要な関数の引数や内部変数などの情報を記録
  - 用途に合わせて収集対象イベントや収集情報（レジスタ内容など）を動的に変更可能

（1）イベントハンドラの動的切替え

（2）情報の採取内容の動的一括切替（**Maskset**切替）

イベントハンドラ：イベントの発生によって実行される関数

Maskset：トレースするイベントと各イベントが使用するハンドラを定義する仕組み

LKST の特徴

- トレース・オーバヘッドの軽減
  - 目的にあわせてトレース量を動的に調整可能
  - トレース未実行時のオーバヘッドを抑える方式を実装

（2）記録情報の種類の動的一括切替（**Maskset切替**）

（3）**Kernel Hooks（GKHI）方式の採用**

GKHI：Generalized Kernel Hooks Interface

LKST の特徴

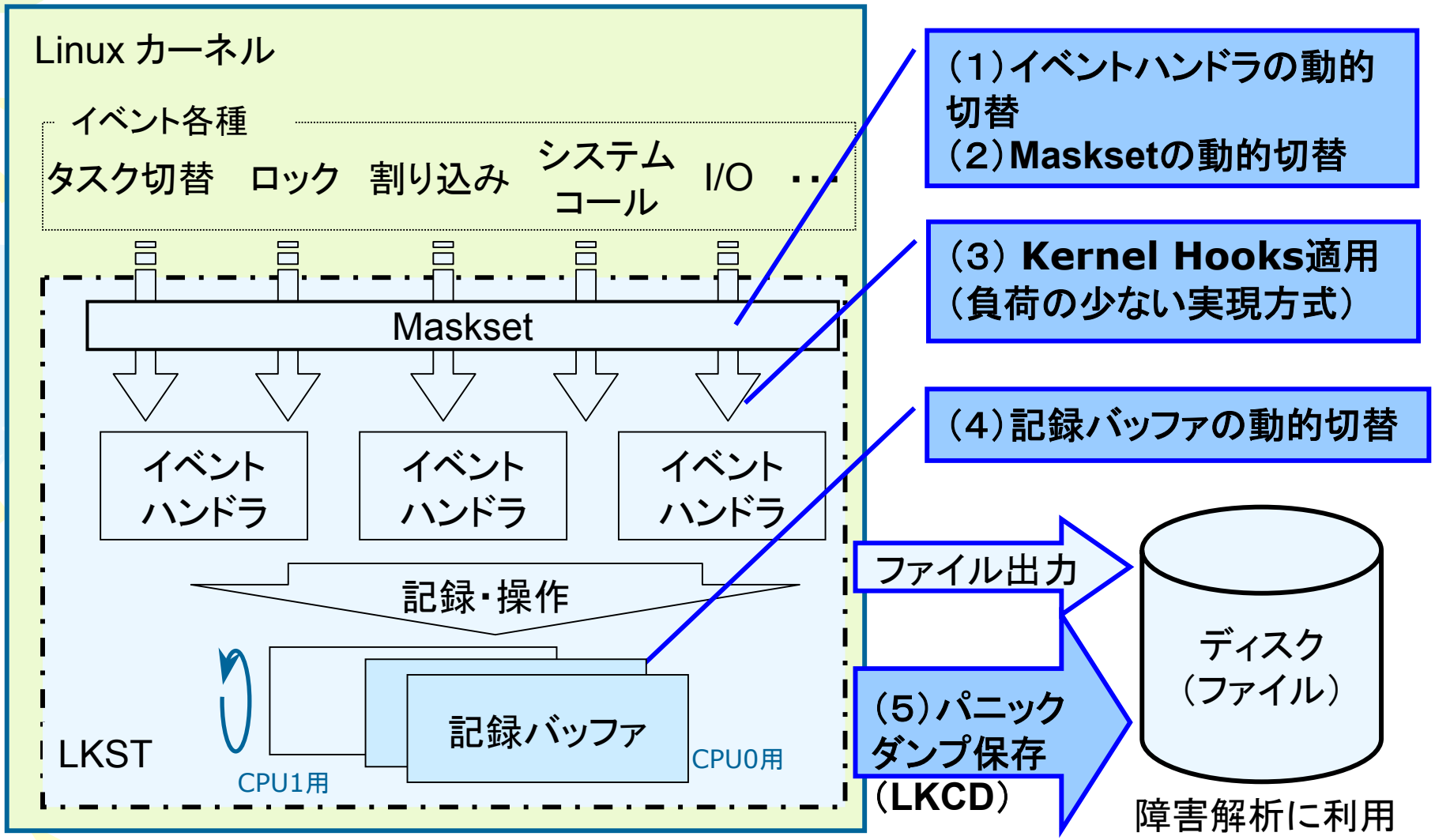
- 確実な情報の回収
  - 膨大な記録の中から必要な情報を回収
  - システムがクラッシュした時に収集していた記録バッファの内容を回収

(4)記録バッファの動的切替

(5)**LKCD**との連携

LKCD：Linux Kernel Crash Dump

LKST の構成



イベント一覧

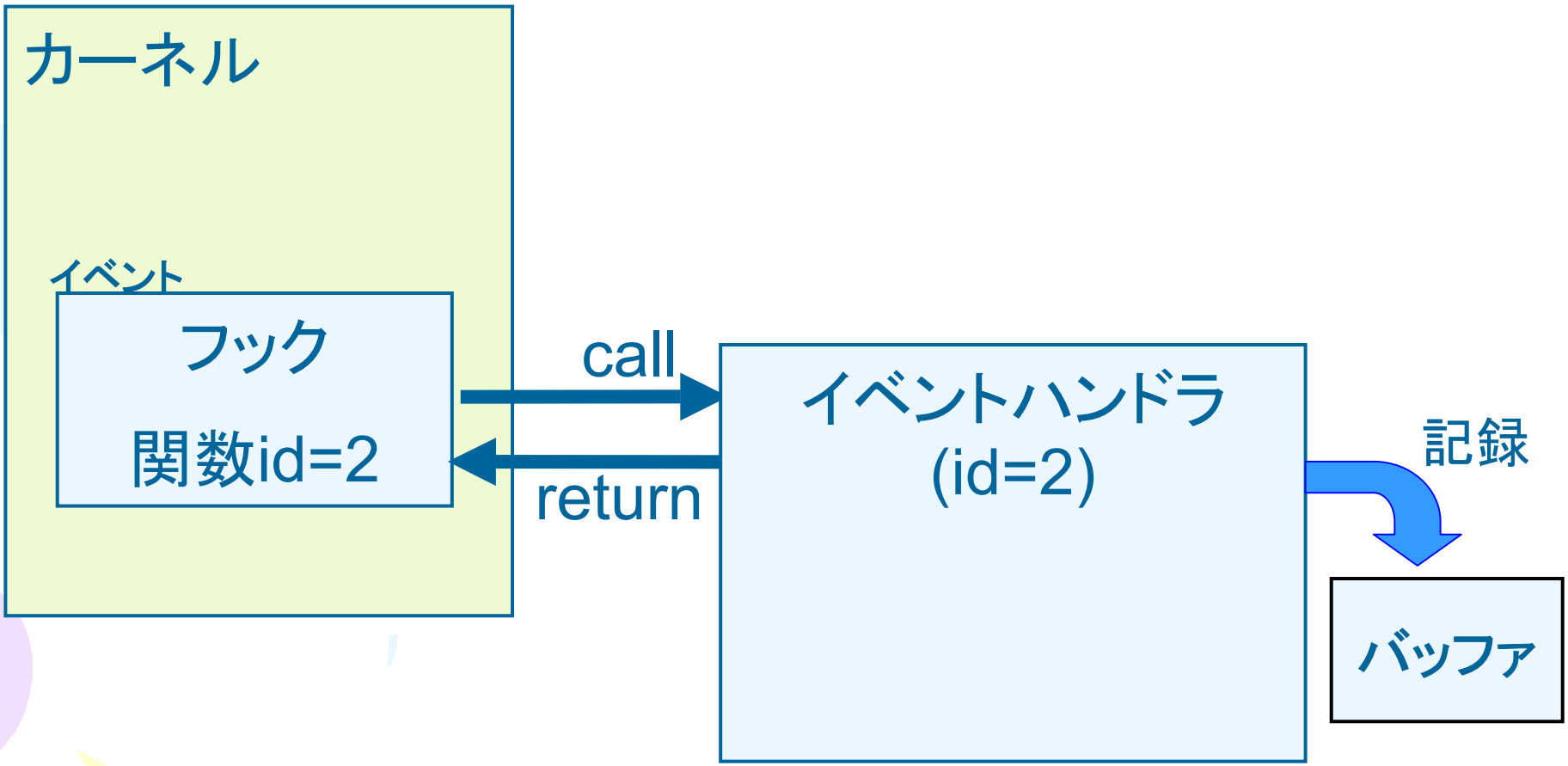
Categoly	イベントの種類
Process・management	PROCESS_CONTEXTSWITCH
	PROCESS_WAKEUP
	PROCESS_SIGSEND
	PROCESS_LTHREADGEN
Interrupts	INT_HARDWARE_ENTRY
	INT_TASKLETHI_ENTRY
	INT_TASKLET_ENTRY
	INT_BH_ENTRY
Exceptions	EXCEPTION_ENTRY
	EXCEPTION_EXIT
System calls	SYSCALL_ENTRY
	SYSCALL_EXIT
Memory・Management	MEM_SWAPOUT
	MEM_SWAPIN
	MEM_DO_NOPAGE
	MEM_DO_WPPAGE
	MEM_WAIT_PAGE
	MEM_GET_FREEPAGE
	MEM_GET_ZEROPAGE
	MEM_FREEPAGE
	MEM_VMALLOC
	MEM_VFREE
	MEM_CACHE_CREATE
	MEM_CACHE_ALLOC
	MEM_MALLOC
	MEM_CACHE_FREE
	MEM_FREE

Categoly	イベントの種類
Filesystems	FS_DEVRW
	FS_DEVEND
	FS_BUFBUSY
Networking	NET_PKTSEND
	NET_PKTSENDI
	NET_PKTRECV
	NET_PKTRECVI
	NET_SOCKETIF
SysV IPC	SYSV_IPC
Locks	LK_SPINLOCK
	LK_SPINTRYLOCK
	LK_SPINUNLOCK
	LK_WRLOCK
	LK_WRTRYLOCK
	LK_WRUNLOCK
	LK_RDLOCK
	LK_RDUNLOCK
Timer	TIMER_RUN
	TIMER_ADD
	TIMER_MOD
	TIMER_DEL
	TIMER_DEL_SYNC
Others	O_PORTIN
	O_PORTOUT
	O_PANIC
	O_PRINTK

イベントハンドラの動的登録・切替

- ユーザ作成のカーネルモジュールをイベントハンドラとして登録可能
- 各イベント毎に使用するイベントハンドラを登録可能
- イベントハンドラは任意の時点で切替え可能

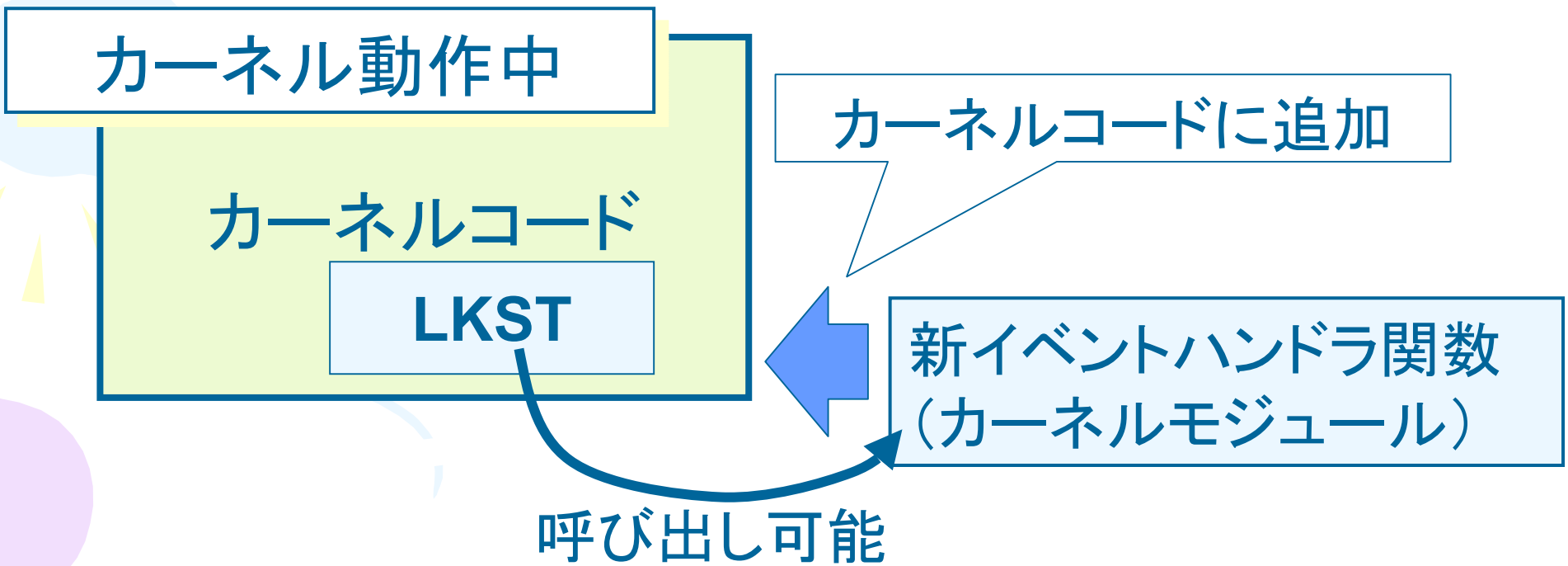
イベントハンドラの呼び出し



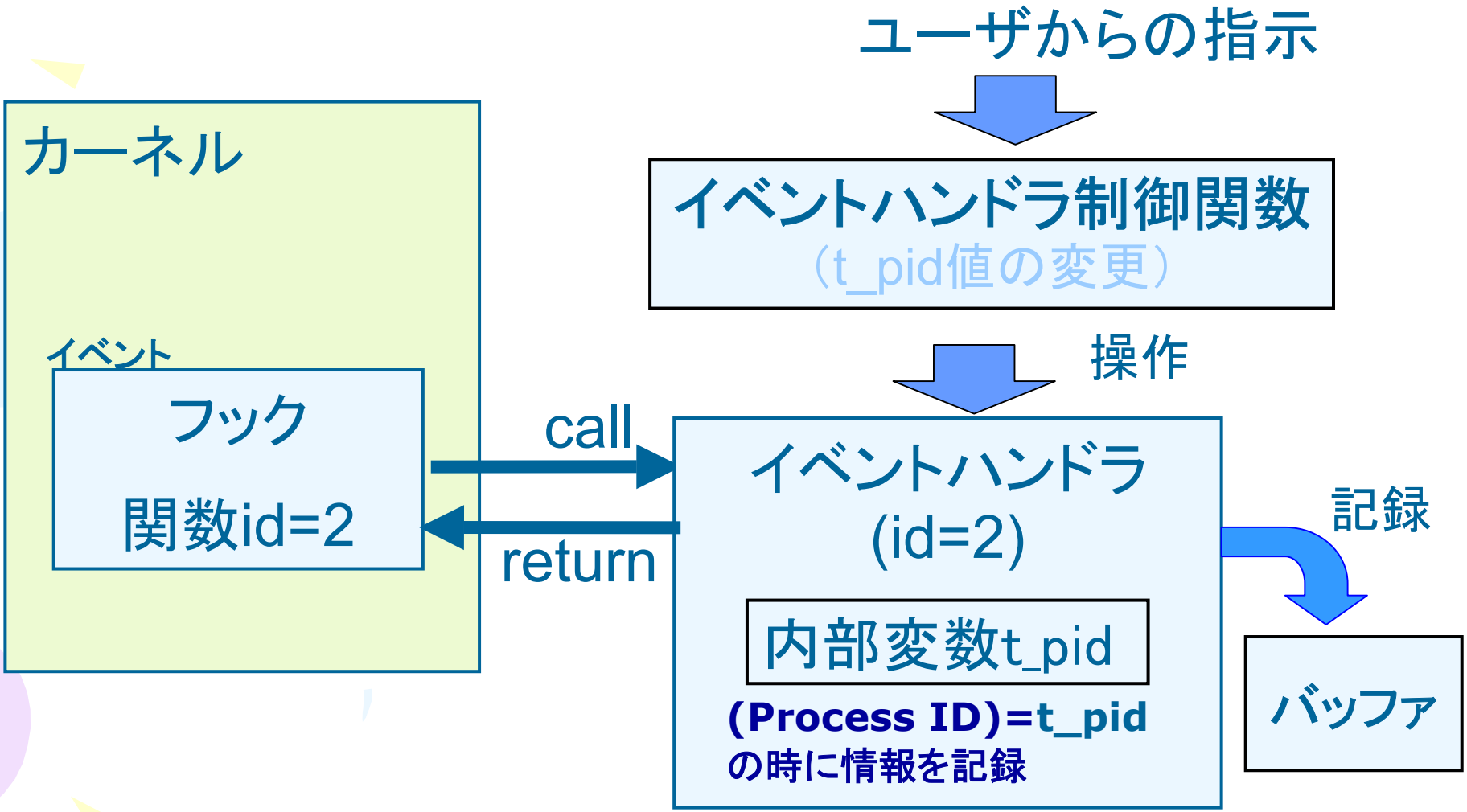
LKST特徴機能(1)

イベントハンドラの動的登録・切替

- イベントハンドラの新規追加



イベントハンドラ制御



イベントハンドラの動的登録・切替

- ユーザ作成のカーネルモジュールをイベントハンドラとして登録可能
- 各イベントに使用するイベントハンドラを登録可能
- イベントハンドラは任意の時点で切替え可能

◆その時点での目的に適した情報の採取方法へ、カーネルのリコンパイル無しに追加・切替可能

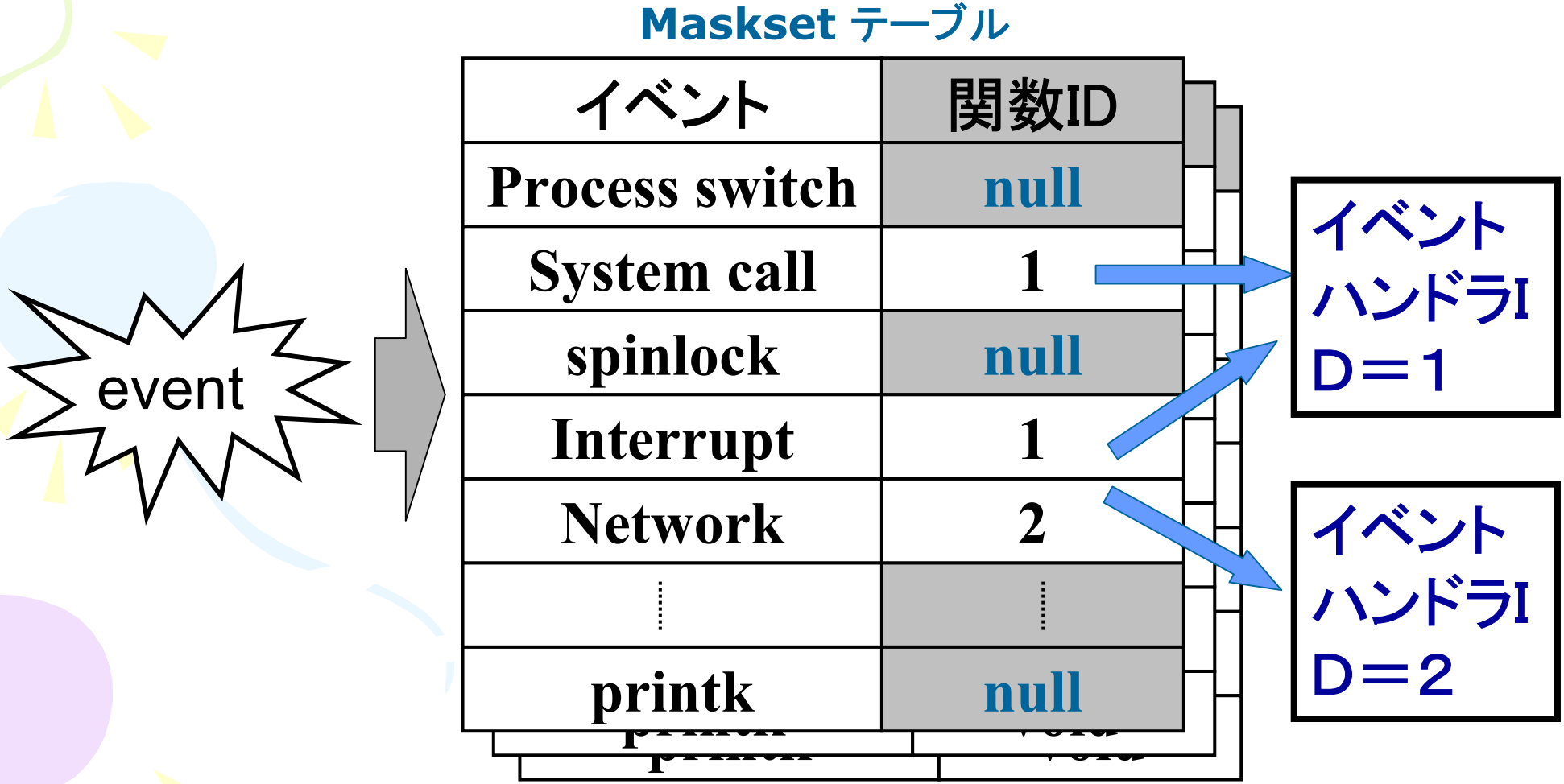
⇒多様なシステムに対応できる拡張性を実現

LKST特徴機能(2)

記録情報の種類の動的一括切替

- Maskset(トレースするイベントと各イベントが使用するハンドラを定義する仕組み)の実現
- Masksetは複数登録でき、かつ動的に切替え可能

Maskset機構



LKST特徴機能(2)

記録情報の種類の動的一括切替

- Maskset(トレースするイベントと各イベントが使用するハンドラを定義する仕組み)の実現
- Masksetは複数登録でき、かつ動的に切替え可能

◆その時点の目的にあう種類のイベントの情報収集へ、動的に変更可能

◆必要がない時には情報量を抑えるなど、必要に応じて情報量を変更し、運用でオーバヘッドを削減可能

LKST

Kernel Hooksの適用

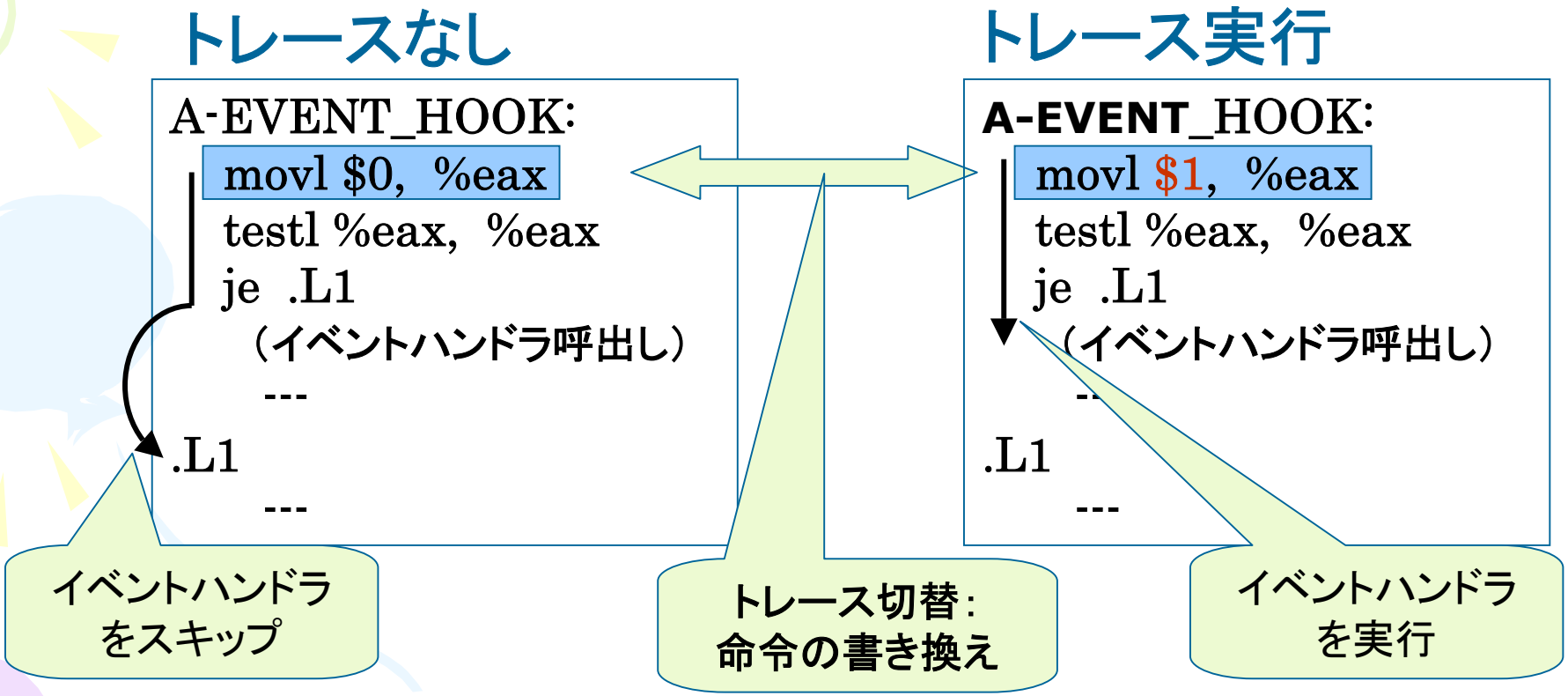
- Kernel Hooks(旧GKHI)とは
  - ・関数をカーネル内のフックと結びつける仕組み
  - ・フックで関数を呼び出さない場合の処理オーバヘッドが少ない
  - ・オープンソースで提供されている
- 連携方法

  イベントハンドラをトレースイベントの記録関数として登録する仕組みに利用

GKHI:Generalized Kernel Hooks Interface

LKST特徴機能(3)

イベントハンドラ実行の判定方式



⇒トレース未実行時のオーバヘッドがほとんど無し

LKST

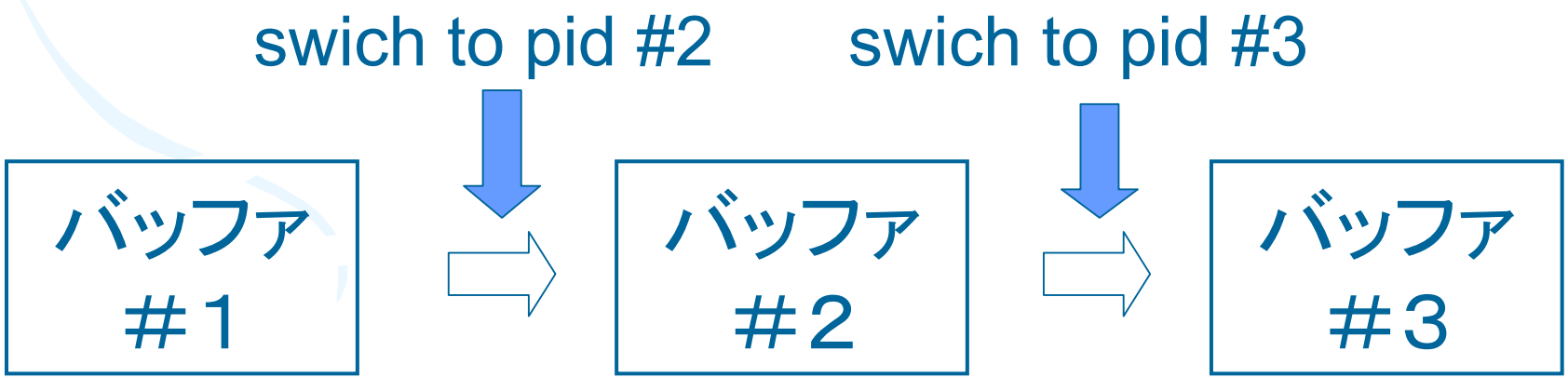
記録バッファの動的切替

- ユーザの指定による動的な切替えも可能
- CPU毎に複数の記録バッファの作成が可能

- **指定したイベントのトレース情報を特定バッファに保存**
- **Lockless処理により、SMP構成でのオーバヘッド小**

記録バッファの動的切替例

- 特定プロセスのイベント情報を保存したい
  - 複数の記録バッファを作成
  - 「プロセススイッチ」イベントに対して記録バッファを切り替えるイベントハンドラを選択



LKCDとの連携

●LKCDとは

システムがクラッシュした時に、メモリの内容をディスク等にダンプし、再起動後に解析できるようにする仕組み。オープンソースで提供されている。

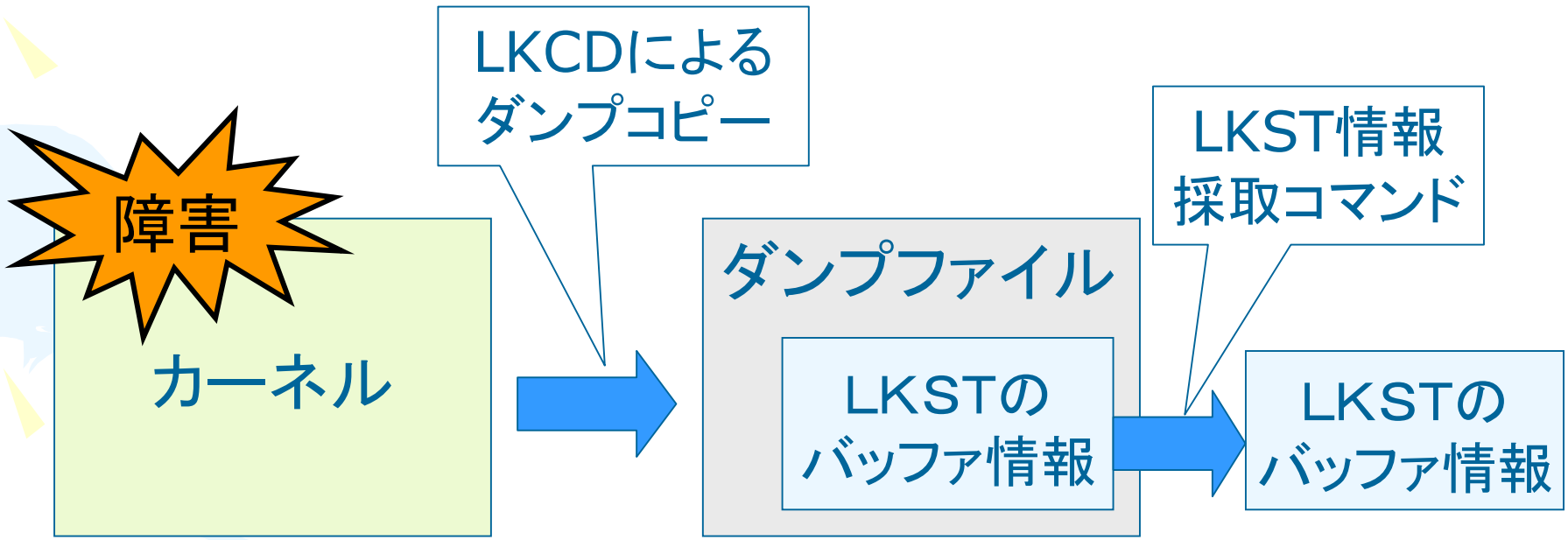
●連携方法

LKCDにLKSTの情報を扱う機能を追加:再起動後に得たダンプからLKSTが記録した情報を採取できるように拡張

LKCD:Linux Kernel Crash Dump

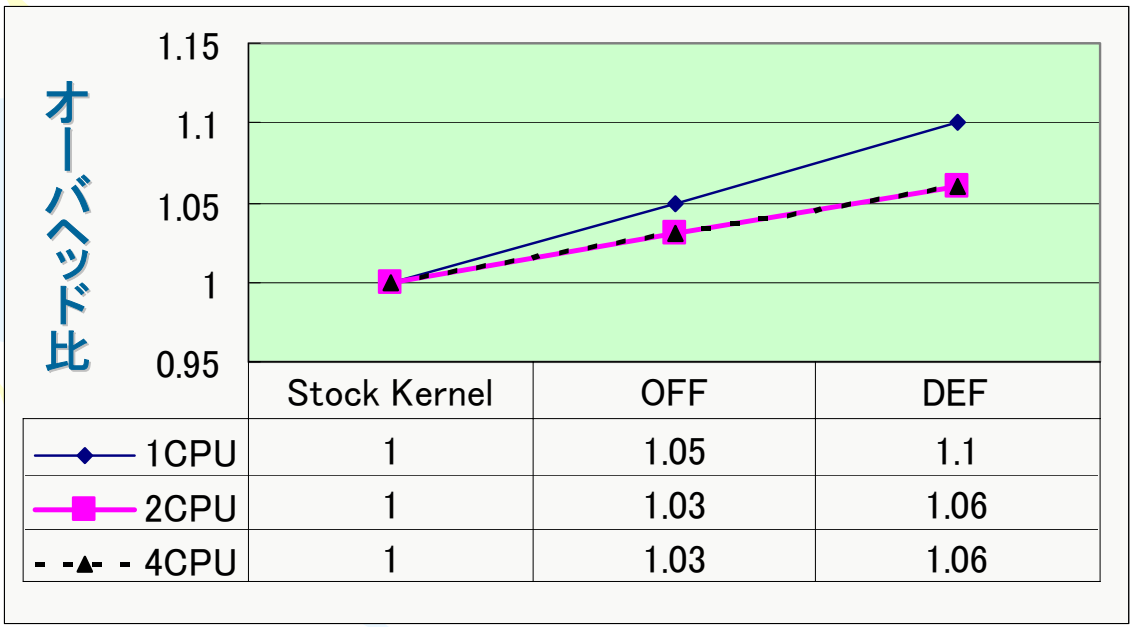
LKST

LKCDとの連携



LKST性能評価

●WebBench 3.0



	Stock Kernel	OFF	DEF
1CPU	1	1.05	1.1
2CPU	1	1.03	1.06
4CPU	1	1.03	1.06

横軸：実行条件

・Stock Kernel：
LKSTコードを組み込んでいないカーネル

・**OFF**：
LKSTを組み込んでているが、トレースを取っていない

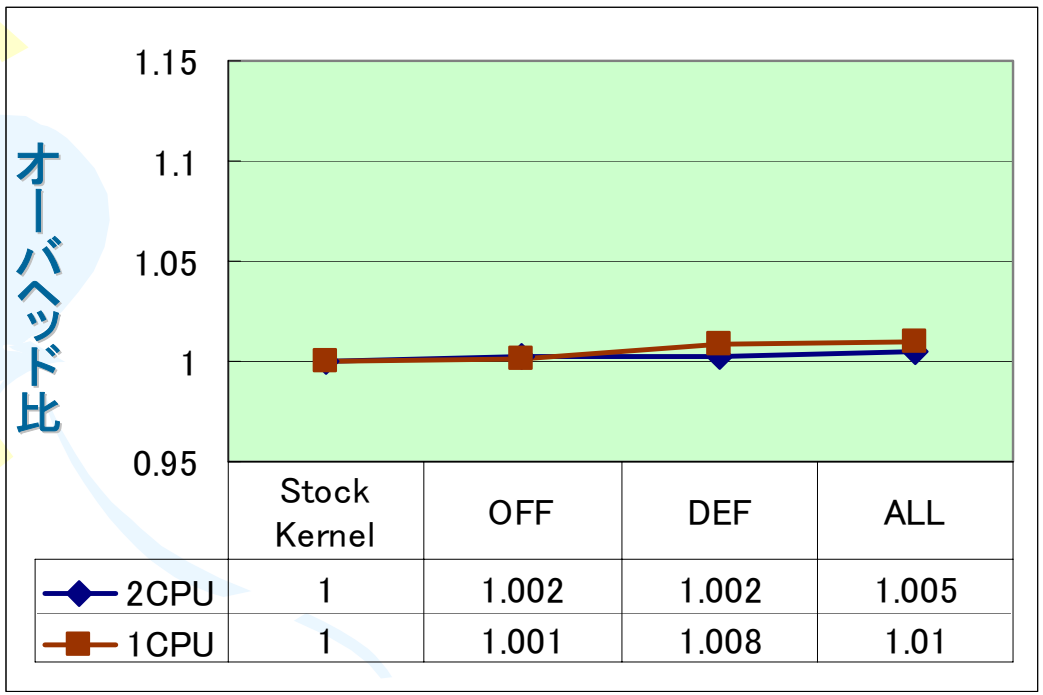
・**DEF**：
ロック関係以外のイベントのトレース実行

測定機器構成：**Pentium3 Xeon 700MHz＊2, 4CPU**
メモリ：1GB
クライアント数＝28台

富士通(株)殿　提供

LKST性能評価

- Kernel recompile



	Stock Kernel	OFF	DEF	ALL
2CPU	1	1.002	1.002	1.005
1CPU	1	1.001	1.008	1.01

測定機器構成：Pentium3 Xeon 900MHz＊2CPU
メモリ：1GB、キャッシュ：2MB

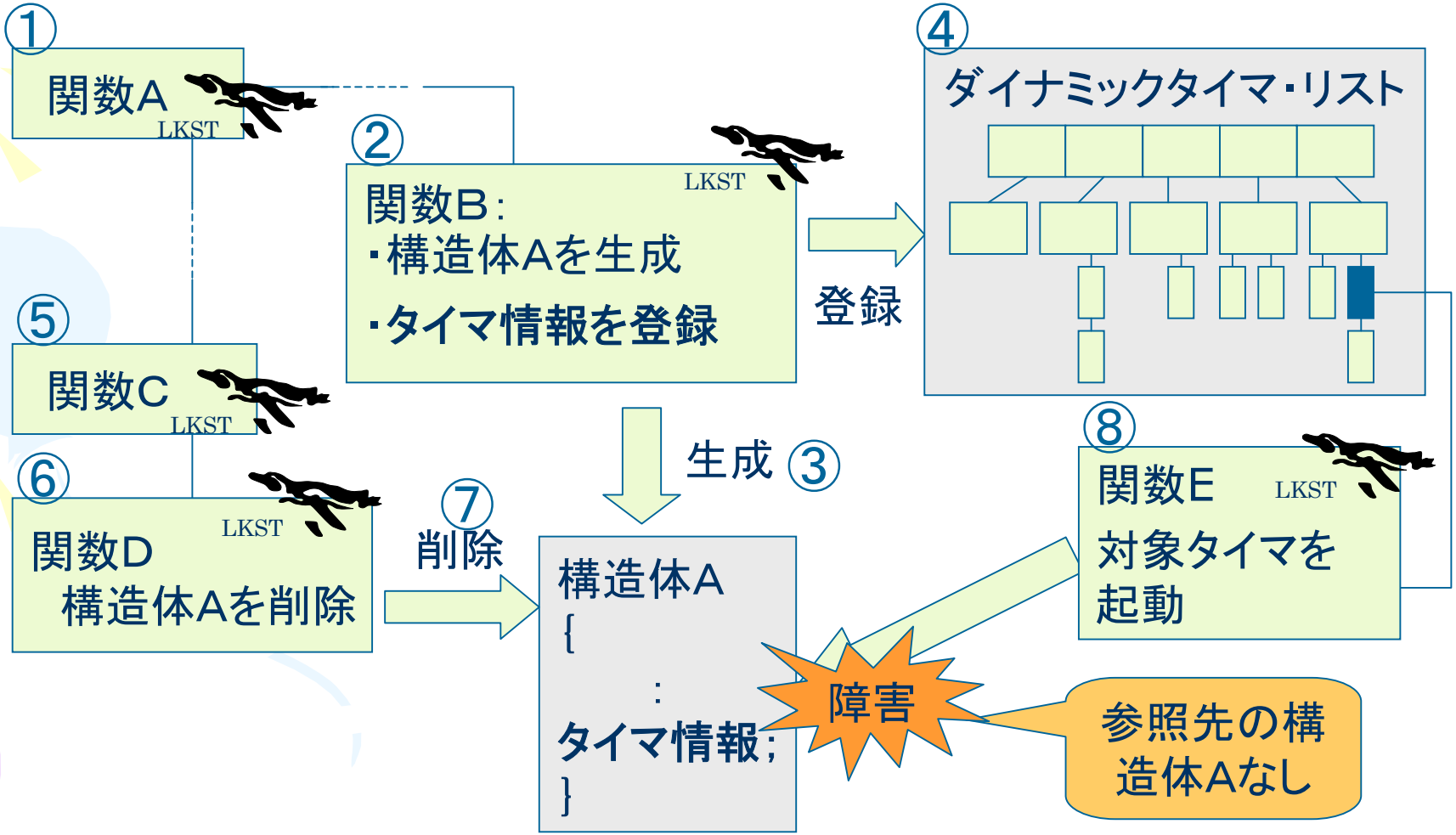
横軸：実行条件

- StockKernel:
  LKSTコードを組み込んでいないカーネル
- **OFF**:
  LKSTを組み込んでているが、トレースを取っていない
- DEF:
  ロック関係以外のイベントのトレース実行
- ALL:
  全イベントトレース実行
  （イベント数約60）

使用例:メモリアクセス障害

- **問題:**
  - タイマが起動する処理で不正メモリを参照，カーネルがダウン
- **調査方法:**
  - 全タイマ操作，および上記タイマを含む構造体Aを操作する全関数において，状態情報を記録

使用例：メモリアクセス障害



使用例：メモリアクセス障害

- **問題:**
  - タイマが起動する処理で不正メモリを参照，カーネルがダウン
- **調査方法:**
  - 全タイマ操作，および上記タイマを含む構造体Aを操作する全関数において，状態情報を記録
- **調査結果:**
  - 構造体Aを消去した後，構造体Aに含まれていたタイマが起動しているのをトレース上で確認

    ⇒構造体Aの消去処理のタイミングを変更

LKST開発状況

●リリース

- 2001/12 ベータ版
- 2002/03 v0.9
- 2002/04 v1.0
- 2002/05 v1.1
- 2002/06 v1.2
- **2002/09 v1.3**

●実装状況

- Stock Kernel ：2.4系（2.4.16, 2.4.17は実証済）
- Linux for AP8000（日立メインフレーム）
- MIRACLE LINUX Standard Edition Version 2.1
- Red Hat Linux Advanced Server 2.1 powered by MIRACLE

今後の予定

- LKST機能強化
  - オーバヘッドのさらなる削減
  - フックポイントの自由な設定
  - 性能評価やセキュリティ強化ツール等への拡張
- 他ツールとの連携
  - Dprobes/Kprobes
  - Event logging

など

LKSTへのアクセス

●ソースコード公開ホームページ

- SourceForge ：
  https://sourceforge.net/projects/lkst/
  http://sourceforge.jp/projects/lkst/
- 日立 ： http://oss.hitachi.co.jp

●メーリングリスト

- lkst-users@lists.sourceforge.jp
- lkst-users@lists.sourceforge.net
- lkst-develop@lists.sourceforge.jp
- lkst-develop@lists.sourceforge.net
- oss@ml.itg.hitachi.co.jp