

SpringOSの設計理念

知念

北陸先端科学技術大学院大学 高信頼ネットワークイノベーションセンター
Dependable Network Innovation Center,
Japan Advanced Institute of Science and Technology

アウトライン

- StarBED Project 背景
- SpringOS 設計理念
- SpringOS 設計
- 10年の知見、当初想定との比較

話者紹介

知念 賢一 *Ken-ichi Chinen*

JAIST DNIC、NICT STC の両方に所属

1990年代: WWW の研究

- 大規模 WWW サーバ
- 先読み代理サーバ

2000年代: ネットワーク実験の研究 (今日の話題)

- ネットワーク実験記述言語
- 各種制御サーバ

< StarBED Project > 背景、当時の状況

- 各種ネットワーク技術の検証が困難
 - ◇ スケーラビリティ（特にインターネット）
- 機材の制限が大きかった
 - ◇ 「NS2 で動かして安心する」
 - ◇ 「手近な数台で動かして安心する」
- インターネットシミュレーション設備として StarBED は始まった

StarBED 設置前夜詳細は提案者の篠田先生へ

< StarBED Project > 歴史

- 2001年まで、篠田研究室でアイデアを練る
- 2002年4月、通信・放送機構「北陸IT研究開発支援センター」開所
- 2003年8月、知念着任、9月より本格始動
 - ◇ 2003年10月、資源管理サーバ等を作り始める
 - ◇ (2004年頃か) SpringOSと名付けられる
- 2006年6月、NICTの「北陸リサーチセンター」として再スタート
- 2011年4月、「北陸StarBED技術センター」改称

< SpringOS > おおまかな経緯

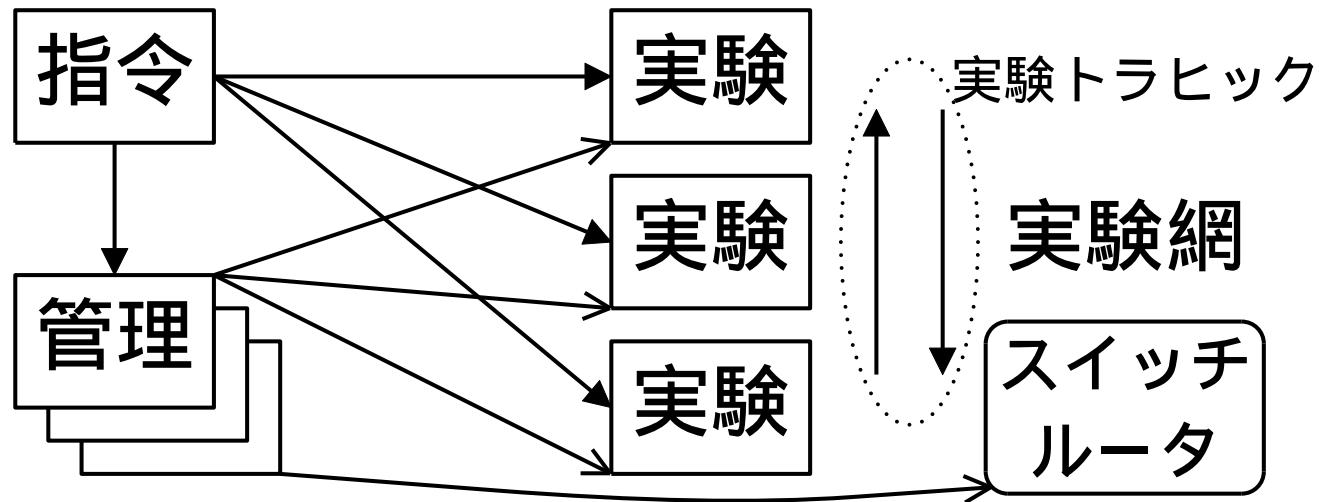
- はじめにいくつかアイディアがあった
知念参加は 2003年夏から
- 欠けている / 薄いアイディアを補強
- 知念の趣味も加えた
- おおむね実装した
 - ◇ 未実現: 技術不足、想像不足
 - ◇ 廃れた: 実状乖離
- 想定外をいくつか追加

理念

- 多数ユーザが同時に実験可能
- 現実的な構成
 - (極力) 実際に使われているハードやソフト
- ホスト設定から実験実施まで一括処理・自動実行
 - ◇ 機材選別、OSやアプリのインストール
 - ◇ 機材電源投入、OS実行
 - ◇ ネットワーク設定
 - ◇ (実験者持ち込み) アプリ実行

設計

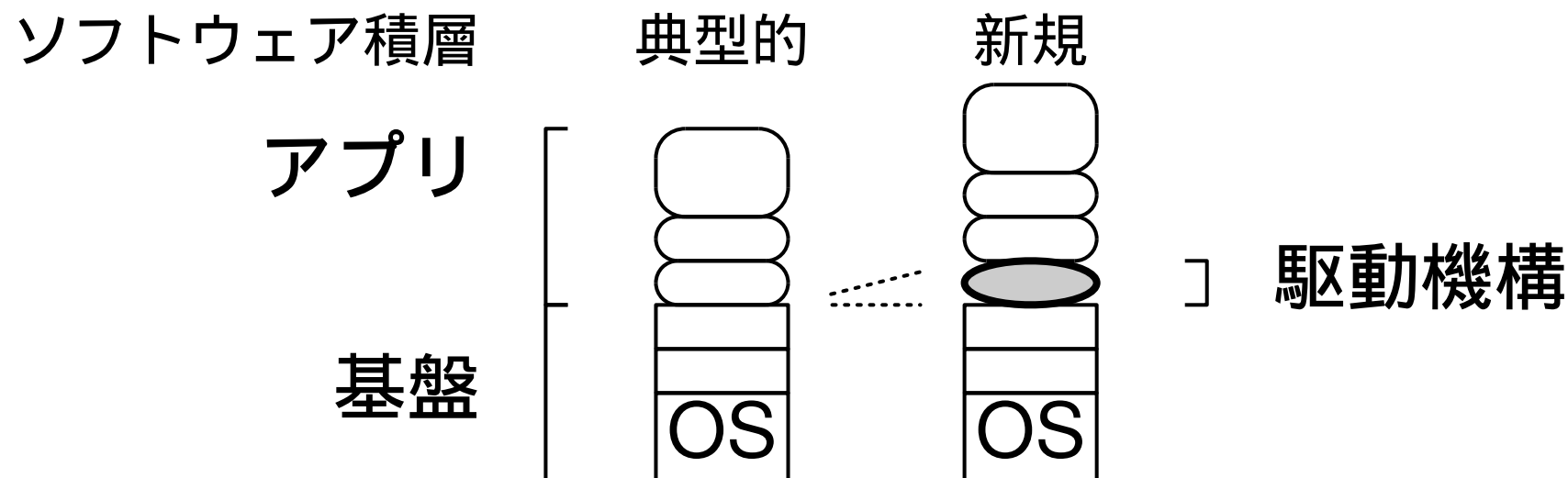
- アプリ実行担当: 駆動機構
 - ◇ ミドルウェア、分散実行型スクリプト言語
- 電源、ネットワーク等各種設定担当: 管理サーバ



ミドルウェアとしての実現

基盤ソフト（OS等）やアプリを極力邪魔しない

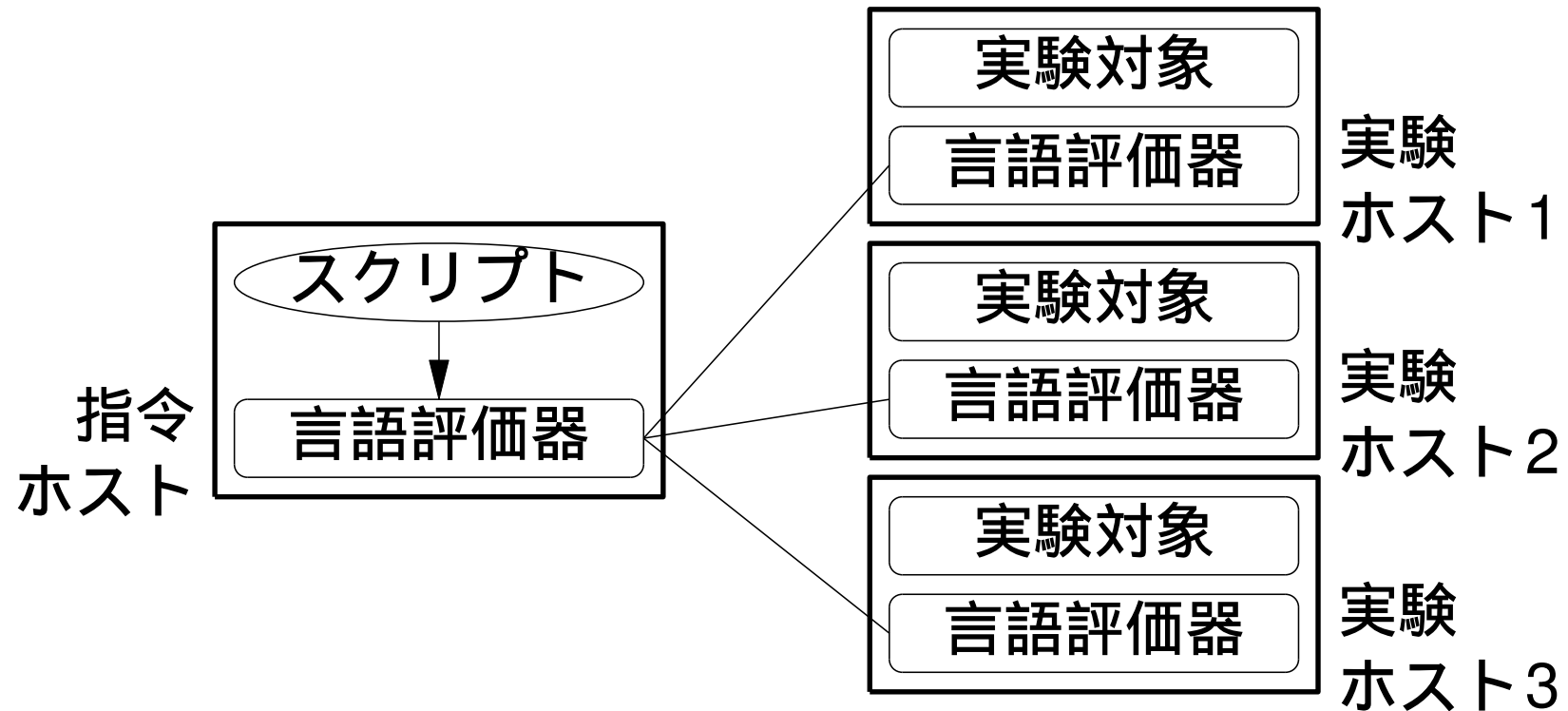
- ソフトの起動・停止
- 他ホストとのメッセージ交換



スクリプト言語 — システム名 Kuroyuri

- ネットワーク実験をスクリプト言語で実現
- 実験実体「ノード」のクラス化
同じ仕様や役割をもつノードをまとめる
- 全体と各ノードの挙動（処理）を分けて記述
処理の流れを「シナリオ」と呼ぶ
 - ◇ 記述ファイルは一つ、ノードに自動配布・実行
- 外部プログラム起動で実験対象アプリ実行
- メッセージでデータやタイミングを交換

スクリプト言語 — システム名 Kuroyuri (*cont.*)



スクリプト言語化の意義

- 実験者の実験プログラムを有機的に結合
- 実験手順をファイルとして固定化、再現性が高い
- タイミング等、パラメータを調整容易
- 変数や関数を用いて、即値を極力排除可能
- ノード数も任意に変更可能

*

*

*

- 実験手順を一つのファイルに凝縮
 - ◇ 細切れファイル散逸防止、バージョン管理容易

TIPS

- 評価器の引数で変数値を代入可能
 - ◇ スクリプトを書き換えずにパラメータ変更可能
 - ◇ デフォルト値を代入可能
- ノードへ変数を転送 (export) 可能
- ネットワーク参加ノードにIPアドレス自動付加
- IPアドレス操作
 - ◇ ネットアドレス抽出、アドレス加算
- 明示すれば、ホストとノードの束縛指定可能

K言語記述例

機材選出
シナリオ
外部プログラム起動
送受信、バリア同期

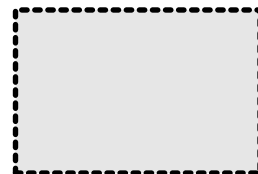
管理サーバ

- 資源管理 (erm)
 - ◇ 資源の属性を保持、公開、検索
 - ▷ 物理的な資源: PC
 - ▷ 論理的な資源: VLAN
 - ◇ 資源の排他利用を制御
- 実験ホスト電源管理 (pwmg)
 - ◇ 古くは SNMP、最近は IPMI
- 実験ホスト起動管理 (dman, kiyomitsu)
 - ◇ TFTP ディレクトリ操作で PXE 挙動制御

管理サーバ (*cont.*)

- 実験ネットワーク管理 (swmg)
 - ◇ マルチベンダー
 - ▷ Brocade, Juniper, Cisco, Dlink, その他
 - ◇ CLI でスイッチ制御
 - ▷ zeroconf, NETCONF は期待しない
 - ◇ 操作を抽象化 // 最大公約数的アプローチ
 - ▷ VLAN 参加 / 脱退、ポート活性 / 不活性

	当初想定内	想定外
廃れた	<div style="border: 1px dashed black; background-color: #e0e0e0; padding: 5px;"> 機材選択 </div>	KVM操作
済	<div style="border: 1px dashed black; background-color: #e0e0e0; padding: 5px;"> 一括実行、シェル ソフトインストール </div>	<div style="border: 1px dashed black; background-color: #e0e0e0; padding: 5px;"> スイッチ制御サーバ化 </div>
着手	静的VM対応	動的VM対応 リンクエミュレーション
未着手	L3操作	クラウドソフト連携 ネットワーク視覚化



= 常用機能

変化・対応

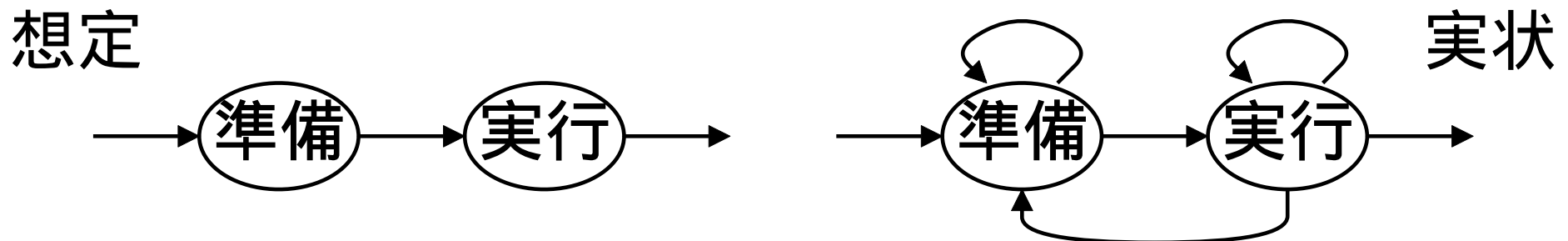
StarBED 運用

- SpringOS を中心に StarBED 運用
 - ◇ 当初は JAIST は 1 へビーユーザ
- StarBED 技術員充実につき独立運用
 - ◇ 技術員が開発したツールも登場
- 予約によって機材が決まる
機材選定は実質なくなった
- 管理領域から利用者の排除
スイッチ制御を管理サーバに移行

変化・対応 (*cont.*)

利用方法、その他

- 一括実施は少数派
 - ◇ 細かな繰り返しが多い
 - ◇ 準備と実行のフェイズがはっきり分かれる



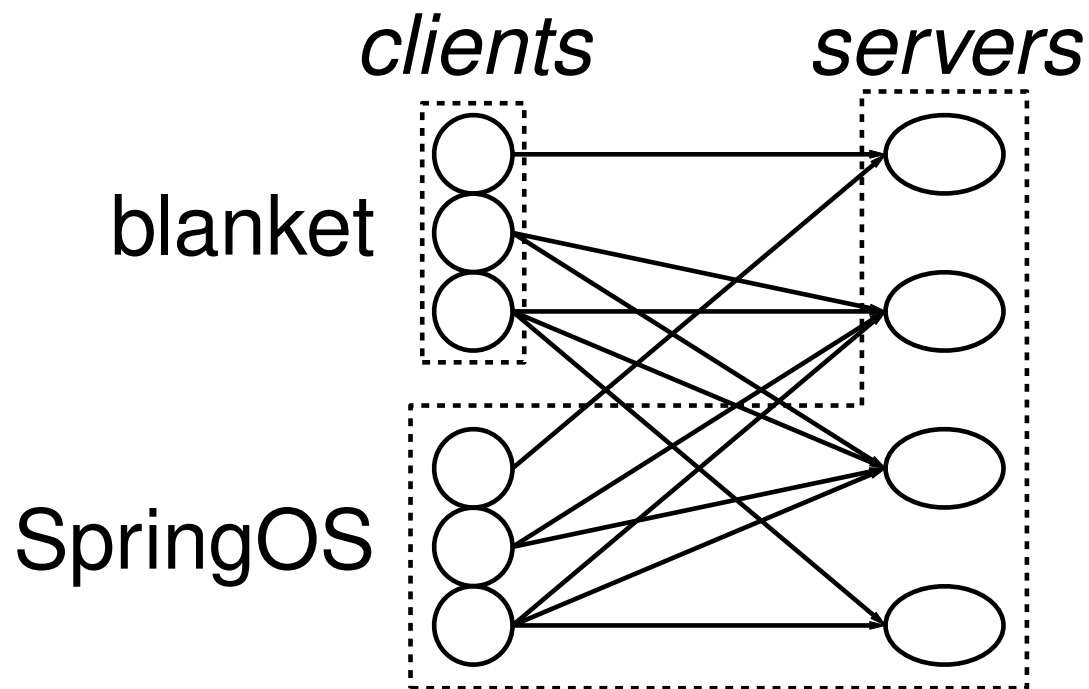
- もう一つのツールキットも登場; blanket
 - ◇ 準備フェイズが充実

変化・対応 (*cont.*)

- GPT や大サイズブートローダの普及
MBR 以降の領域も保存・復元
- 関連施設に各種スイッチ導入
順次対応

blanket: 新クライアント群

- 新たなユーザ側ツール
- サーバは SpringOS そのまま



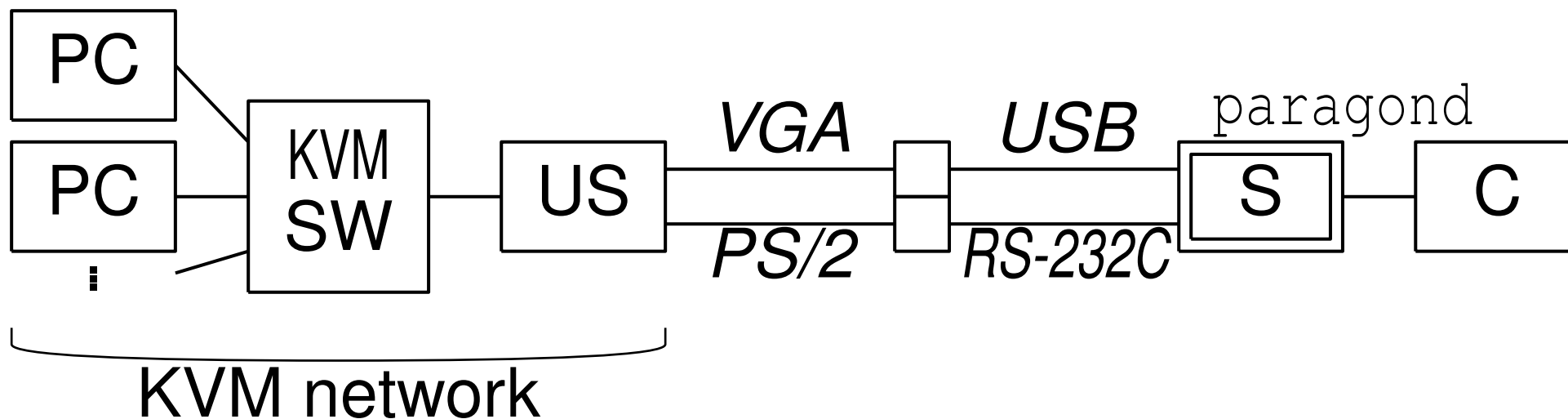
構想内で未着手

- L3 レベルの自動化
ルーティング等の詳細未定

知念の想像力の外

お蔵入り: KVM制御ソフト paragond

- 多数の PC を巡回監視
- 画面出力を解釈したフィードバックも可能
- BIOS 設定も画策していた



KVM 装置使用機会激減で開発停止

将来への展望

『技術が代っても自動実行・全体制御は必要』

近い将来 — ひきつづき開発

- テストベッド連携にも対応

遠い将来 — 応用分野拡大

- クラウドの時代でも
- IOT テストベッド

SpringOS 貢献者（敬称略）

命名: 篠田

基本設計: 篠田、宮地、三角、知念

もしかしたら、篠田研究室先達のアイデアも含まれている可能性あり

詳細設計: 知念

実装: 知念、磯崎

デバッグ、テスト: 宮地、中川、中井

コメント、その他: 三輪、利用者のみなさま