

HPCI共用ストレージ モニタリング環境および可視化情報提供環境

Gfarm Workshop 2025
2025/12/17(wed) @Miyako-jima



理化学研究所 計算科学研究センター
金山 秀智

What is HPCI in Japan ?

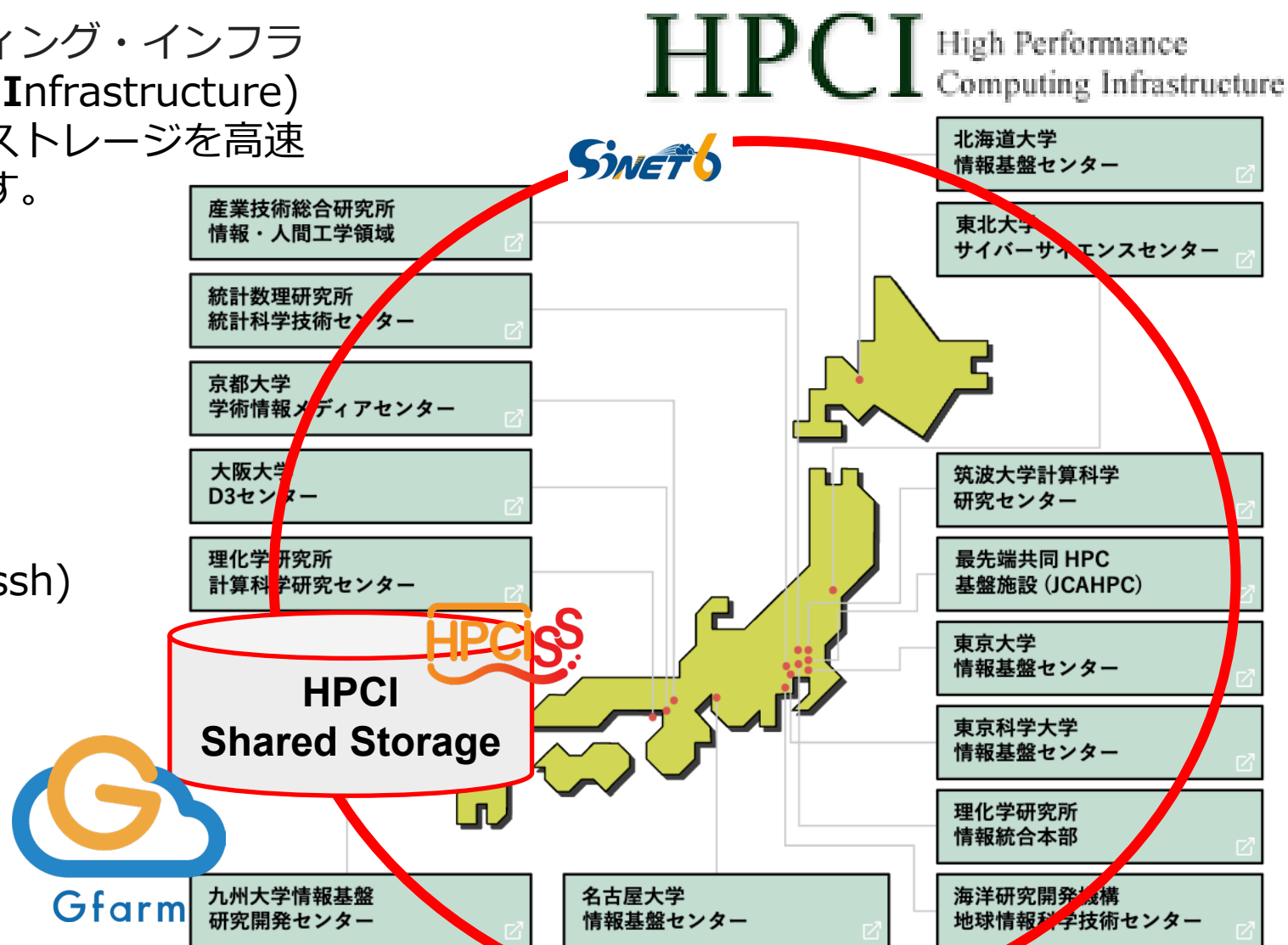
革新的ハイパフォーマンス・コンピューティング・インフラ
(HPCI = **H**igh **P**erformance **C**omputing **I**nfrastructure)
国内の大学や研究機関の計算機システムやストレージを高速ネットワークで結んだ共用計算環境基盤です。

Provide to service

- ・ 課題選定
- ・ アカウント管理
- ・ シングルサインオンサポート
 - ・ Web: Shibboleth
 - ・ システムログイン: OAuth認証(& hpcissh)

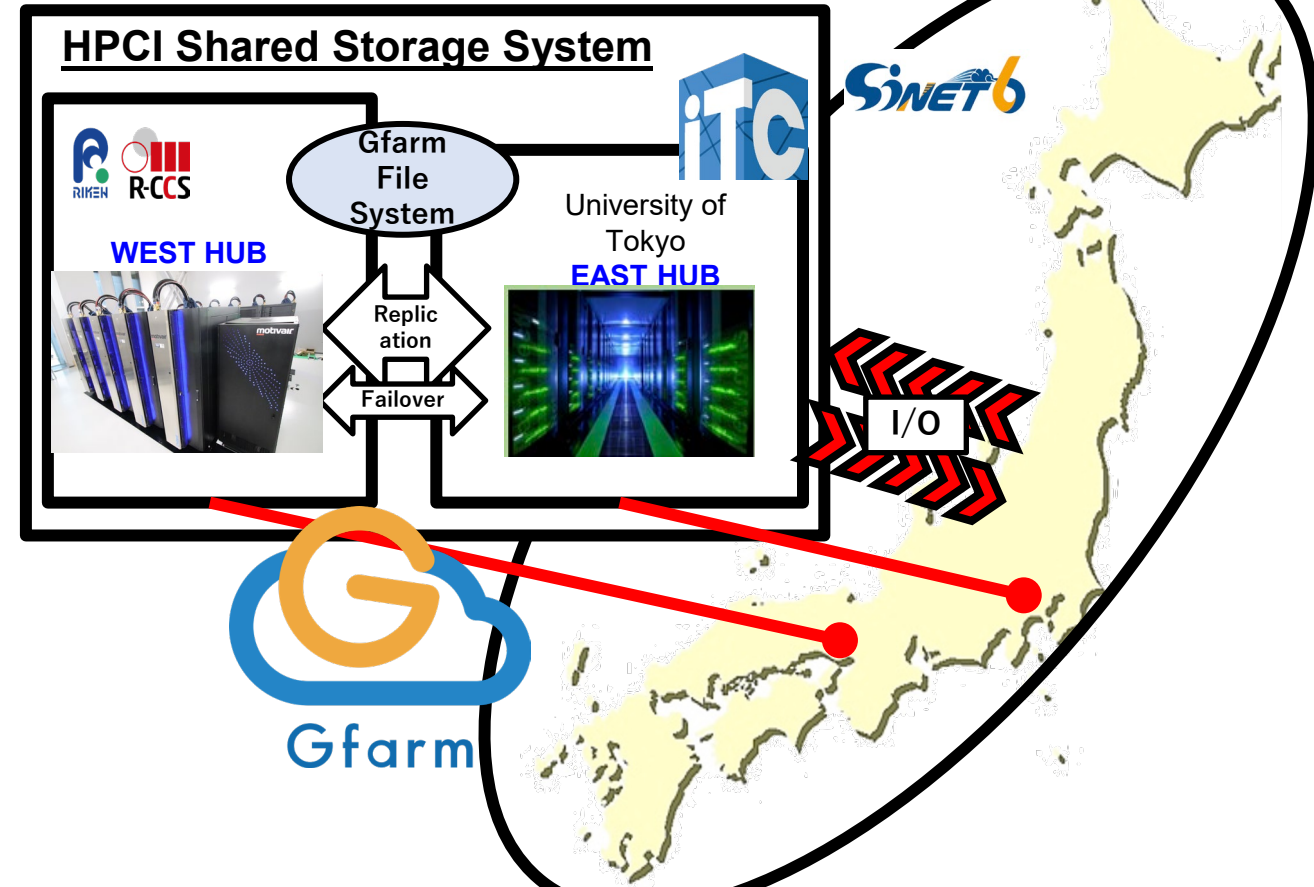
Provide to Infrastructure

- ・ ヘルプデスクサポート - RIST
- ・ ネットワークストレージ
→ HPCI Shared Storage System



What is HPCI Shared Storage?

- **目的:**
 - スーパーコンピュータ間(計算資源)間のデータ共有
 - 研究データの保存
 - 研究データの公開(公的データの利活用)
- **運用機関**
 - 東京大学
 - 理化学研究所(R-CCS)
- **基幹ソフトウェア**
 - Gfarm File system
 - <https://github.com/oss-tsukuba/gfarm>
- **特徴**
 - HPCI資源のクライアントから並列転送によるデータI/Oが可能
 - 2拠点間でのデータレプリケーションによるディザスタリカバリ
 - 高可用性(99%の稼働率 – 運用停止/アクセス不可は1%未満)
 - 高速転送可能なネットワークストレージ(200Gbps以上での接続)



Gen	Periods	Capacity(Logical)
1st	FY2012 – FY2018	10PB → 15PB
2nd	FY2018 - FY2024	45PB → 50PB
3rd	FY2025 -	95PB+

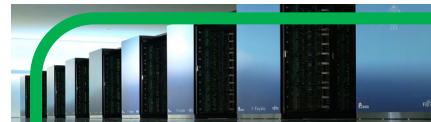
R-CCS HPCI共用ストレージ 第3期システム



HPCI共用ストレージシステム西拠点はR-CCS(神戸) 計算機棟に設置されています。

- ・第2期システムは計算機棟1階、
第3期システムは計算機棟3階に構築。
- ・ネットワーク等は既存のものを流用せず。
- ・冷却用の空調機(水冷式空調機(リアドア))も合わせて調達。

3F: 計算機室 – 富岳・HPCI共用ストレージ(3rd Gen)



HPCISS

空冷

水冷

水冷式空調

2F: 空調機械室



1F: 計算機室 – 富岳(ストレージ)・HPCI共用ストレージ(2nd Gen)
量子コンピュータ/AIスパコン



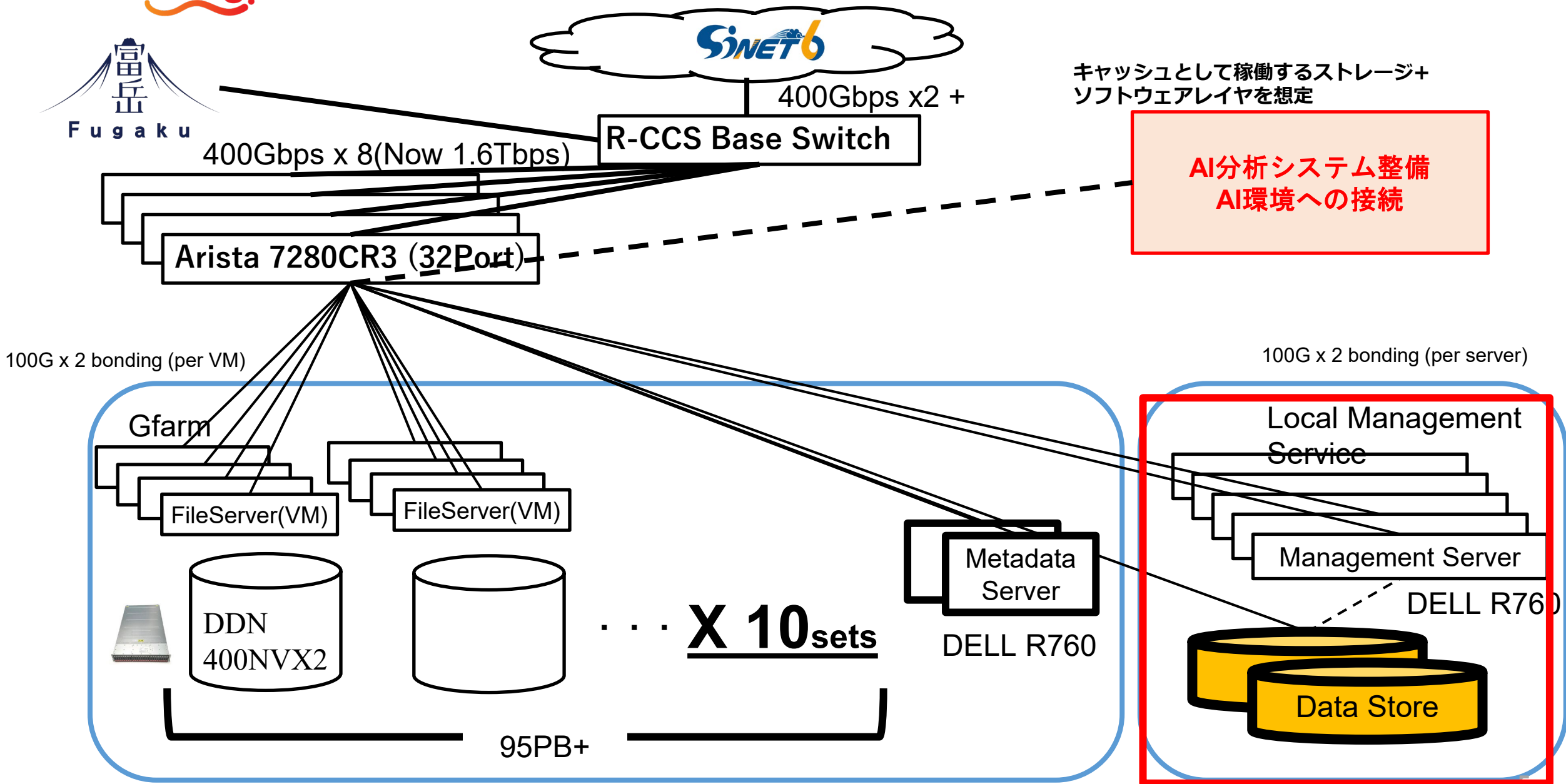
HPCISS

空冷

B1F: 空調機械室



System Configuration(3rd Generation/R-CCS)



HPCI共用ストレージのモニタリング環境

現状は以下の通り監視環境が別れている。HPCI共用ストレージ(特にGfarm)では、メトリクスだけではなくログ情報からも監視情報が提供されているので、本来はログとメトリクスを集約して可視化・アラート発砲を調整する必要があるなどの問題を抱えている。

目的	(1) 従来の監視(Zabbix監視)	(2) SRE Monitoring (サービス監視 ユーザ情報提供)	(3) ログ収集
概要	<ul style="list-style-type: none"> 主にGfarmPluginを用いた監視 フェイルオーバーの自動実行 障害アラートの発砲 	<ul style="list-style-type: none"> サービス全体の障害把握(東西間のアラート情報を[一部]集約) サービス自体の稼働状況などから障害レベル(※)に合わせたアラート発砲 電話サービスによる高レベル障害発生時のアラート通知 ユーザへの情報提供 メトリクス収集 	<ul style="list-style-type: none"> Gfarmログ情報の集約 ログ可視化
主な利用ソフトウェア	Zabbix(可視化・解析・アラート・収集・保存) Gfarm Zabbix Plugin(収集・アラート) ShellScript(ログ・メトリクス収集)	Grafana(可視化) Twillio(電話サービス) AlertManager(アラート) Prometheus(収集・保存) exporter(メトリクス収集)	Kibana(可視化) ElasticSearch(検索/分析) Logstash(パーサー)

なぜこうなっているのか？

- 2018年からサービスレベル向けのモニタリング環境を整備
 - SRE Monitoringがトレンド(Grafana + Prometheus)
 - 富岳や他の環境でも取り入れ・検討開始
 - Zabbixが高負荷/エラー対応が負担になる傾向
 - Zabbixによる可視化が当時は貧弱
- 結果サービスレベルで問題や情報を集約するためのモニタリング環境を整備

一方現在はモニタリングは「流行っていない」「あたりまえ」に…

現状LLM等のAI連携に向かっているので今後大きく環境が変更する可能性があります。

(OSS GrafanaではLLM Pluginなどが活用できないので、
LLM連携が行いやすい環境などに移行しないとまずそう！！)

→現状OSS環境などでは良さそうな方法があまり見つからず。情報が欲しいです。

SREとMonitoringおさらい

- 2010年代に流行ったSite Reliability Engineering(サイト信頼性エンジニアリング)
(Googleのシステム運用のベストプラクティス – 2003年から提唱)

- 従来は、異常検知が最終目標
- SREでは、開発への貢献が最終目標
- 本来はGfarmへの貢献をすることを期待
→ **これがなかなかできていない。ごめんなさい。**
→ 本来運用者、管理者、モニター、開発、調査員等は細分化前提
→ HPCI共用ストレージは“そうっていない…”
- SREの指標は主に以下。
 - SLI (Service Level Indicator):
監視する具体的な指標 (例: リクエスト成功率、レイテンシ)
 - SLO (Service Level Objective):
SLIに対する具体的な目標値
(例: 99.9%のレスポンスタイムが500ms以下)
- ITILに即してHPCI共用ストレージでは
「稼働率(ユーザへのデータアクセス提供)」をSLIとしてSLOを定義
それに対して障害レベルを再構築。
 - 監視も合わせて障害レベルを再定義して、それに乗っ取る形での監視を始めることとた(2018年)
 - このため(微妙に残っているが)、継続的に把握できない/実現できない内容を漠然と「障害や問題」として定義されても対処できない。
→ 例: 検知ができないユーザデータの消失、ユーザアクセスができなかったこと、データが流出したこと
 - これらは障害指針として別軸の話で進める必要がある (これを納得してもらうのがとても難しい)



SREとMonitoringおさらい②

● SRE Monitoringの観点としては

- サービスレベルの定義に合わせたエラー判断が重要
 - 結果アクセス失敗などの情報をエラーとして検知するような調整を開始
 - 内部からの情報集約では、エラー判断が不足
→ ブラックボックス監視および外形監視を取り入れ
- 実際の利用状況の把握が必要



- 「情報の集約と分析・活用」「個々のエラー・障害対応」からの解放が目標
- → 省力化およびGfarmやソフトウェア・ハードウェアへの貢献
- エ → 実エラーへの詳細対応からサービスレベルでの問題対応へ
- → 性能改善や問題点の分析を可能にするための情報集約
- ※ 利用者への情報提供はあくまで副次的な話。
- エラーはあくまで改善に利用

関連サービス 第二期システム→第三期システム①

項目	サービス名	目的	第二期システム	第三期システム	第二期 – 第三期の違い
モニタリング環境	Zabbix + Mysql	Gfarm Zabbix Pluginとの連携しGfarmの主要な障害を報告	1台のサーバの仮想マシンで稼働(週1でbackup実施、ストレージは1台のサーバに紐付いている)	データストアとZabbixサービスを切り離し。データはスナップショット backupと直近1週間分のみを保存などを検討。	DBとサービスを切り離し、backupの高度化とシングルポイントへの対策を行う。 第二期システムでは、DB性能やサービス性能で、監視間隔を上昇させられなかったが、サービスを分割して分割するなども含めて検討。
	Prometheus + Influx DB	メトリクス情報の収集	第二期システム運用中に構築 1台のサーバの仮想マシンで稼働。 Exporter開発なども行い、メトリクス収集を開始。サービスレベルの監視を行い、自動電話サービスなどと連携して障害の速報/24時間監視を自動化	第二期からメトリクスを減らさず、監視間隔の上昇、ebpfなどによる詳細なカーネル情報の収集、水冷装置などの新規ハードウェアの監視、高レベル障害の対応自動化。コンテナ利用を行い冗長化させる。	サーバ性能などで頭打ちになっていた監視間隔を増加させる。情報取得を並列化させるなどで情報取得・DBへの書き込みを並列化・高速化。冗長化。
	Prometheus + Timescale DB	同上(外形監視/ユーザ別情報の収集)	第二期システム運用中に構築 ユーザの個別情報などを収集、セキュリティの観点からPrometheusを直接Grafanaと接続するのではなく、TimeScale DBを噛ませている。	メタデータ性能の向上があり、情報収集頻度が上昇するため、ユーザ情報取得の更新頻度を増加する。コンテナ(仮想マシン)で冗長化させる。	情報収集頻度の増加。冗長化。
	Grafana + MySQL	Prometheusで収集したメトリクスの可視化	第二期システム運用中に構築 1台のサーバの仮想マシンで稼働(シングルポイント) PrometheusやDBアクセス性能で細かいメトリクスの可視化では少し時間を要している	コンテナ(仮想マシン)で冗長化させる。 各種DBアクセス性能および情報収集頻度向上にともない、可視化の詳細化	可視化の詳細化。冗長化。
	Elastic Serach + Kibana + logstash	ログ情報の収集、監視、可視化	第二期システム運用中に構築 各サービス1台ずつの仮想マシンで構築されている。(シングルポイント) 並列実行などがされていないので性能はサーバに依存	冗長化・並列化を行い、ログ収集・可視化情報の出力を改善させる。Gfarmなどのソフトウェアのデバッグログも含めて、リアルタイムで分析する環境をめざす。	分析性能向上、冗長化

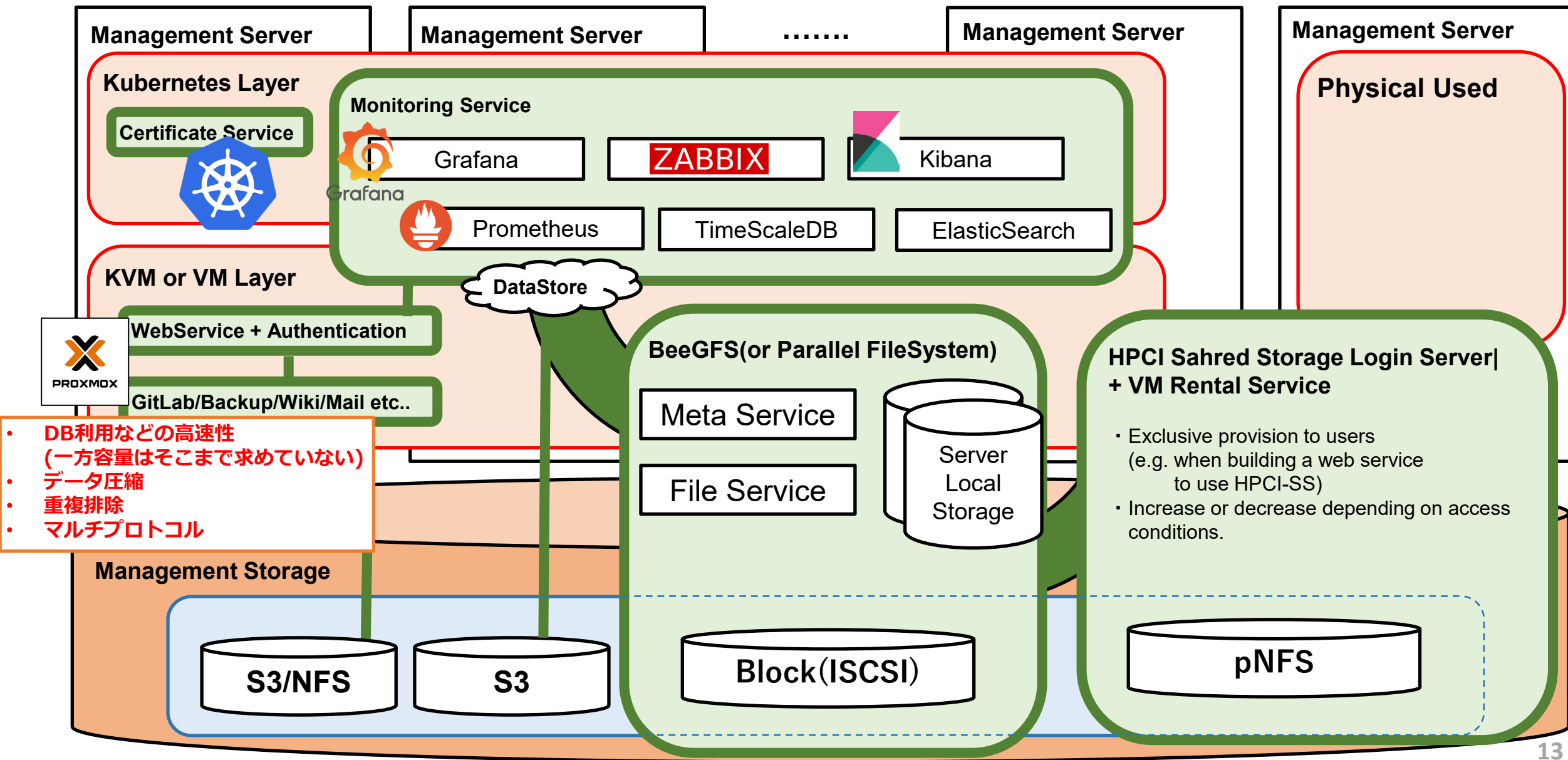
関連サービス 第二期システム→第三期システム②

項目	サービス名	目的	第二期システム	第三期システム	第二期 – 第三期の違い
HPCI共用ストレージ向けログインノードサービス	ログインノード	HPCI共用ストレージのデータ書き込み・読み出しに利用	初期6台で構築、10Gbpsと帯域が不足	100Gbpsで構築予定。 仮想マシンの利用を行い、利用状況に伴い追加などを行うことを検討	利用状況に合わせた提供を検討。
HPCIアカウント管理	Shibboleth IdP + DB(LDAP -> Directory 389)	HPCIアカウントIdPサービス およびPWMサービス	仮想マシン x 2を用いて 異なるサーバ感動して冗長化	DBサービスとIdPサービスを分離、 コンテナ環境で冗長化(もしくは KVM冗長化) DBサービスは管理サーバ向けデータ ストアに設置 PWMサービスを切り離して冗長化	第三機では、冗長性をk8sや Proxmox等のオーケストレーション ツール側で実装することで、 サーバ故障などでも冗長構成が維持 されることになる。 またDBのスナップショットなどを 世代ごとにとることでバックアップ 容量の低減が可能
Webアクセス環境	Apache + Shibboleth SP	SP提供(による認証・認可)および 各種Webサービスのリ バースプロキシ	第二期システム運用中に構築 1台のサーバ上の仮想マシンで稼働 (backupはアップデート時などに取得)	Proxmoxなどでの冗長構成を持たせて backupはスナップショットを活用 し定期的に取得	第三期システムでは、シングルポ イントの解消を目指す
データ公開	Next Cloud	データ公開環境の整備	—	Next Cloudを用いたデータ公開環境 を目指す。一方NextCloud以外での 公開環境や、プラットフォーム連携 などが課題(ユーザ寿長などと合わせ てサービス追加)	新規サービス開始
Globus	Globus 5 -> Globus 5 + 6	データ転送プロトコルの追加 「Gfarmなし」環境との連携	第二期システム運用中に構築 Globus 5系でのgsi 認証/公開鍵認証 での並列転送	Globus 5および6系でのGlobus Platform連携をめざす。	Globus 6サポート (Globus Platform対応)
Syslogサービス	syslog	HPCIログ規定の通り、最低5 年間のログを保存	1台のsyslogサーバに保存	Syslogサービスとデータストアの分 離、冗長化	冗長化
メールサービス	Postfix	ユーザへの自動連絡や障害ア ラートメール向け	1台のメールサーバに保存	メールサービスの冗長化	冗長化

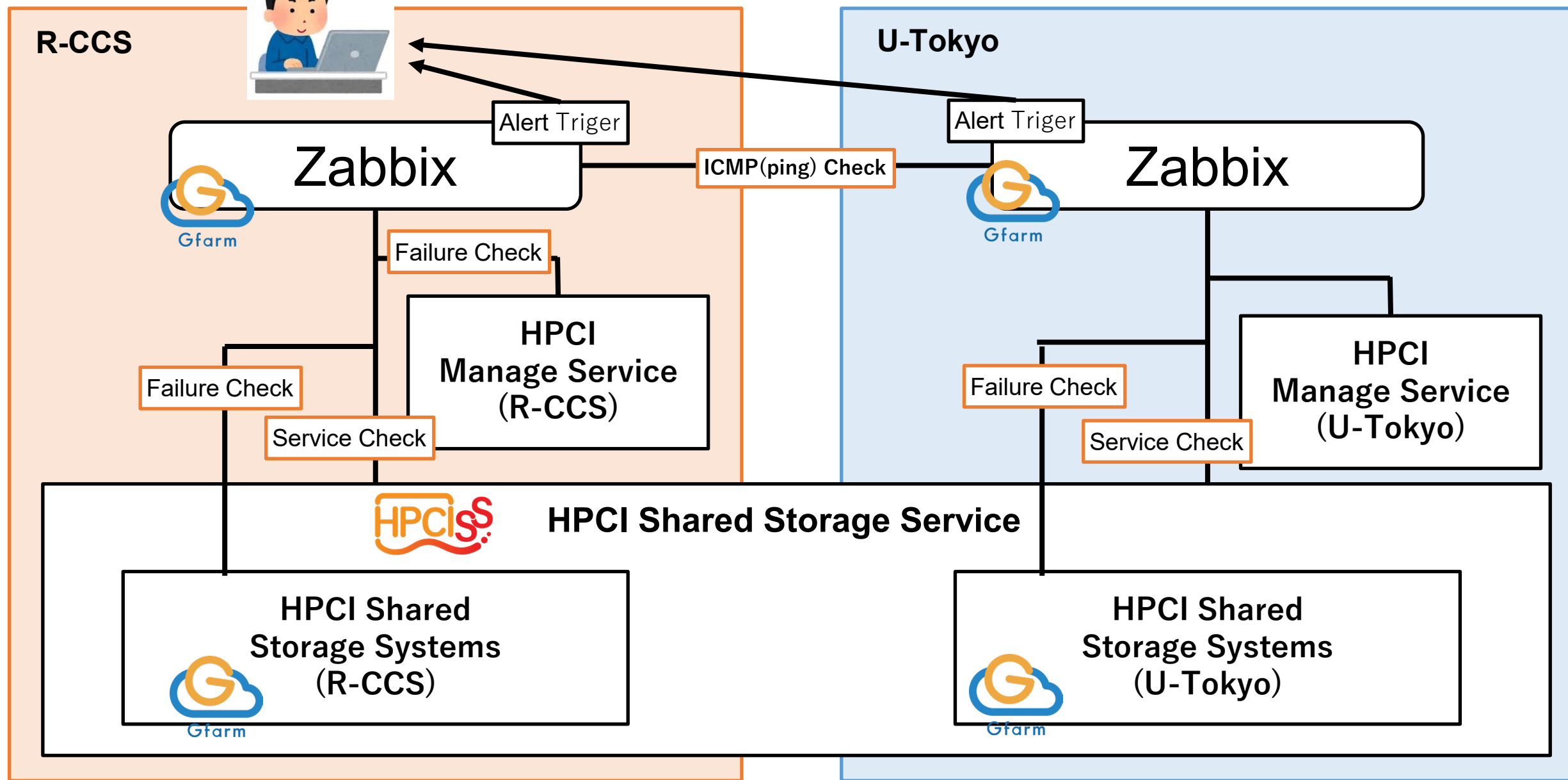
関連サービス 第二期システム→第三期システム③

項目	サービス名	目的	第二期システム	第三期システム	第二期 – 第三期の違い
証明書	EJBCA	HPCI共用ストレージ向け証明書環境	第二期システム運用中に構築(NII証明書終了のため)	データストアデータのbackupの改善(スナップショットへの移行)	冗長化、backup改善
オーケストレーション(自動化)	Ansible / AWX or Ansible Automatic Platform	インストール・設定自動化	Ansibleの導入、Gitリポジトリで管理	GUI(Webベース)環境の導入、環境の冗長化、gitサービスとの連携	GUI化
管理者向けログインノード	SSHD	役務ユーザや外部からの管理者アクセス向け	特段第一期システムから変更無し(SSHDが実行できて、アカウント管理などの実行元として利用)	AIエージェントなどの導入を行うことを検討	AIエージェントの活用(?)
Backup System	—	バックアップを保存	サーバにHDDを複数挿してRAID6を構築して利用	ストレージシステムを利用	IOPSやスナップショットなどの機能、S3、NFSなどにも対応
Wiki	Local Wiki(Pukiwiki/Redmine)	内部連携	R-CCS内部サービスへの移行(Pukiwiki -> Redmine)	R-CCS内部サービスを引き続き利用	—
リポジトリサービス	gitlab	役務管理・連携、役務作業者のコミット	第二期システム運用中に構築Gitlabを運用	第二期システムから継続、シングルポイントの解消を目指す	冗長化
AI/分析	<今後検討>	ログ・メトリクス分析を目指す	—	クラウド環境を用いてクラウド業者が提供中のAIサービスを確認、調査	AI活用

3rd gen systems Local Management Service (constructing now)



(1) Gfarm(ZABBIX)監視環境(従来の監視)



Gfarm(ZABBIX)監視環境(従来の監視)

Gfarm Zabbix PluginおよびGfarmのフェイルオーバー機能の利用を目的としている。

第二期システムのZabbix環境はEOL VersionなのでLocalに閉じている。

また、シングル環境で稼働しており仮想マシン停止や仮想マシンサーバの停止でZabbix監視が停止する状況。

第三機ではZabbix環境をGfarm Zabbix Pluginが対応する最新までアップデートしてk8sで稼働させる予定。

→ <https://github.com/oss-tsukuba/gfarm-zabbix/releases/tag/4.3>

- ・最新版がZabbix 5.0 + CentOS7系の記載なのでZabbix 7系で稼働するか確認が必要。

■ Zabbix監視環境で主に活用している機能および監視内容

主な項目	機能内容	備考
Gfarm Failover機能	マスタgfmdの稼働状況を監視して、障害が発生した場合failoverscriptを実行する。	
Gfarmの<err> <warning>を検知 kernelエラー・I/Oエラーの検知 (基本的なGfarm Zabbix Pluginの機能)	基本的なエラー検知向けの障害報告	ログ監視に集約を目指す予定。 現状のノウハウからLLM連携やユーザへの自動通知なども検討する予定
容量逼迫等のアラート	各種ホストの容量逼迫などを報告	サービスレベル監視でも取得しているので必須ではなくなっている
プロセス状況や外部へのアクセス状況の監視	Pingや稼働プロセス、負荷状況などを監視	同上

お気に入りのグラフ

- ss-08-0.r-ccs.riken.jp: データ移行・二重化: ファイルサーバネットワークINCOMING(ss-*, das-ss*)
- ss-08-0.r-ccs.riken.jp: データ移行・二重化(移行先)ファイルサーバネットワークOUTGOING
- esci-wgfs001.aics.riken.jp: 旧ファイルサーバネットワークINCOMING
- esci-wgfs001.aics.riken.jp: 旧ファイルサーバネットワークOUTGOING
- ss-04-0.r-ccs.riken.jp: Incoming network dropped on bond0
- ms-0.r-ccs.riken.jp: gfmd RSS size

グラフ »

お気に入りのスクリーン

- [R-CCS] SharedStorage
- [R-CCS] データ二重化(Network Bandwidth)

スクリーン »

お気に入りのマップ

- R-CCS 共用ストレージ

マップ »

ウェブ監視

ホストグループ	正常	失敗	不明
---------	----	----	----

ウェブシナリオが設定されていません。

更新時刻: 11:41:13

最新20件の障害

ホスト	問題	最新の変更	経過時間	情報	コメントあり	アクション
private__mng-3.r-ccs.riken.jp	Free disk space is less than 20 % on volume /backup	2025/12/09 13:25:49	7d 22h 15m	いいえ	2	6
gfmng3r02.hpcli.r-ccs.riken.jp	gfstatus replicainfo value is disable	2025/10/24 10:34:13	1m 24d 1h	いいえ	9	6
gfmng3r01.hpcli.r-ccs.riken.jp	gfstatus replicainfo value is disable	2025/10/24 10:30:49	1m 24d 1h	いいえ	9	6
private__mng-0.r-ccs.riken.jp	Free disk space is less than 20 % on volume /data	2025/08/14 21:20:50	4m 4d 14h	いいえ	2	6
sw-mng.mng.hpcli.r-ccs.riken.jp	Free disk space is less than 20% on volume Shared memory (Threshold: 0.8)	2022/05/19 11:01:23	3y 7m 3d	いいえ	1	
sw-mng.mng.hpcli.r-ccs.riken.jp	Free disk space is less than 20% on volume flash: (Threshold: 0.8)	2022/05/19 09:57:22	3y 7m 3d	いいえ	1	
ss-00-0.r-ccs.riken.jp	Problem of gfsd ('/usr/local/gfarm/bi...)	2019/09/04 16:43:57	6y 3m 15d	いいえ		
ss-00-0.r-ccs.riken.jp	Stop checks process gfsd(ss-00-0-1.r-ccs.riken.jp)	2018/10/03 09:23:47	7y 2m 17d	はい (1)		
ss-00-0.r-ccs.riken.jp	Stop checks process gfsd(ss-00-0-0.r-ccs.riken.jp)	2018/10/03 09:23:46	7y 2m 17d	はい (1)		

9件中9件の障害が表示されています

更新時刻: 11:41:13

Zabbixサーバーの状態

パラメータ	値	詳細
Zabbixサーバーの起動	はい	134.160.187.7:10051
ホスト数 (有効/無効/テンプレート)	241	57 / 72 / 112
アイテム数 (有効/無効/取得不可)	15563	14076 / 760 / 727
トリガー数 (有効/無効) [障害/正常]	5649	5323 / 326 [9 / 5314]
ユーザー数 (オンライン)	38	2
1秒あたりの監視項目数(Zabbixサーバーの要求パフォーマンス)	166.81	-

更新時刻: 11:41:16

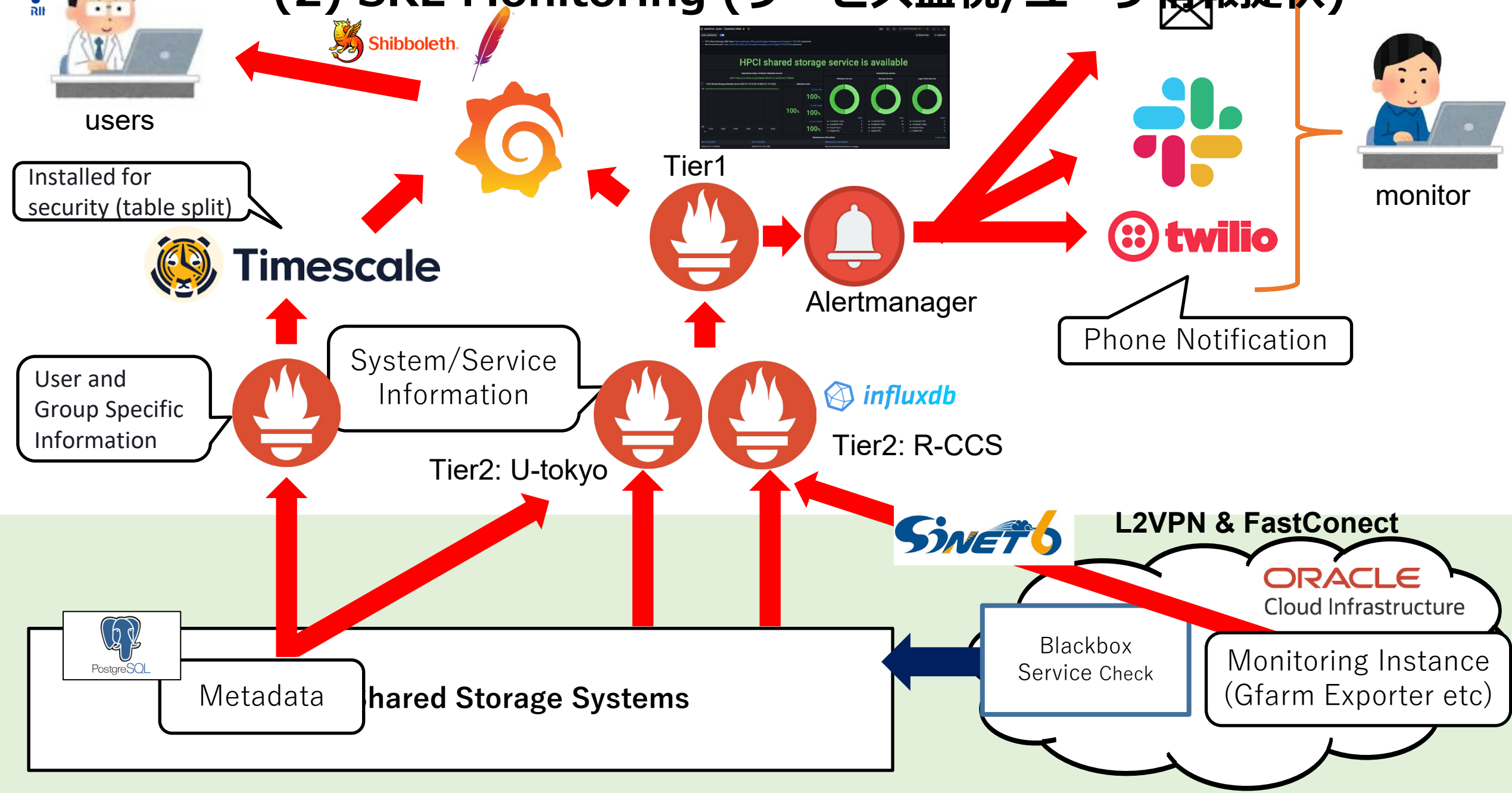
ホストステータス

システムステータス

ホストグループ	致命的なエラー
Linux servers	0
Virtual machines	0
[AseanGfarm] File Serveer	0
[AseanGfarm] Metadata Server	0
[east-hub] Zabbix	0
[west-hub] HPCI	0
[west-hub] LOGIN	0
[west-hub] MANAGEMENT	0
[west-hub] MANGSERVER(PRIVATE)	0
[west-hub] MDS	0
[west-hub] MNGSERVER	0
[west-hub] SS	0
[west-hub] Switch	0
[west-hub] Zabbix	0
[ログインノード運用・保守] EMG-HPCSQL	0
[夜間休日サポート] EMG-II	0
[終業時状態確認] EMG-HPCSQL	0

更新時刻: 11:41:13

(2) SRE Monitoring (サービス監視/ユーザ情報提供)



(2) SRE Monitoring (サービス監視/ユーザ情報提供)

内部的にはさらに詳細な情報を取得し、サービスレベルに合わせたアラートを提供

取得・監視項目	内容	備考
サーバ・ストレージシステム毎の詳細情報	node_exporter等で取得した情報	
ネットワーク情報	snmp_exporter等で取得した情報 (ネットワーク流用やエラーカウントなども)	
メタデータベース状況	postgresql_exporterで取得した情報	
詳細容量情報	各ユーザ・課題毎の詳細な容量 ファイルサイズ別容量・ファイル数 アクセスがないファイル数や容量 レプリカ数情報 レプリカ複製状況	Gfarm メタデータベースから詳細な情報を取得
サービスレベル監視	HPCI共用ストレージの運用においてレベル 分けしている障害レベル別のアラート発砲 を実施(例: R-CCSのファイルサーバが全て アクセスできない状態で東大のファイル サーバ1台が故障 → アクセスできないファ イルが発生)	主にOCIからのblack box監視を元を実施。
ディスク障害対応に向けたメトリクス取得	SMARTやエラーカウントの収集から予防交 換などによる障害前の対応を重視	第二期システムの保守なし運転の実現 QSFPのロット障害の発見 障害傾向に向けたベンダとの議論

サービス監視/ユーザ情報提供

ユーザ情報については、Gfarm APIだけではなくメタデータベースからSQLで情報を取得して情報集約を行っていき、**メリットはスレーブメタを使うことでモニタリングのメタデータに与える負荷を削減できること。**

基本的に以下のテーブルの集計を行うことで概ねの利用情報が取得可能。

今後メタデータベースの拡張情報などを取り扱う場合はさらに他のテーブルを参照する必要あり。

Gfarm APIでは User毎/Group毎の情報は取得できる。
一方で、Groupに所属するUser 別(Group and Userの条件)の利用情報は取得できない
※ Directory Quotaなどを使えばAPIでも取得できます。

Cold Dataであるか、いつ書き込まれたファイル化、読み出しが行われているかなどの情報を収集するために利用

レプリカ作成遅延などが起きていないか、レプリカ作成の進捗、余剰レプリカの状況などを取得
(データ移行の進捗取得等でも活躍)

	Description
	Inode番号
	世代番号
	ハードリンク数
size	レプリカサイズ
mode	ファイルモード(rwx)
username	所有ユーザ名
groupname	所有グループ名
a,m,c}timesec(nsec)	Atime/mtime/ctime秒(+ナノ秒)
inumber	Inode番号
hostname	レプリカの保存先のgfsdホスト名

サービス監視/ユーザ情報提供

GarmのMetadata連携を行う場合はxmlattr情報を用いる予定。
 Gfarm APIを元にして集計するつもりだが、今後増加量が増えるようなら負荷などを考えて
 スレーブメタデータに対して情報集計を行う予定。

DB	Table	Column	Description
Gfarm	xmlattr	inumber	Inode番号
		attrname	XML attrのアトリビュート名
		attrvalue	XML attrのアトリビュート

```

gfarm=> SELECT
  COUNT(DISTINCT i.inumber) AS inode_count,
  SUM(i.size)              AS logical_bytes
FROM inode i
JOIN xmlattr x ON x.inumber = i.inumber WHERE
i.groupname = 'hp120273' and i.username = 'hpci002329'
and attrname = 'test';
inode_count | logical_bytes
-----+-----
135 | 13946060805
(1 行)
  
```

gfarm=>

inumber	attrname	attrvalue
210417956	test	<test_metadata> +
		<write_user>kaneyama</write_user> +
		<write_data>2025/09/18</write_data> +
		<descripts>試験用のテストファイル</descripts> +
		</test_metadata> +

ちょっとGrafanaを捨てたい気分

Grafana OSSの制約がきつい

- ・ SAML認証未対応
- ・ LLM利用連携が弱い(ってかない)
- ・ セキュリティが弱い(SQLインジェクションなどができてしまう)

情報公開する際にユーザ別の詳細情報などを流すときなどで制限を食らっている

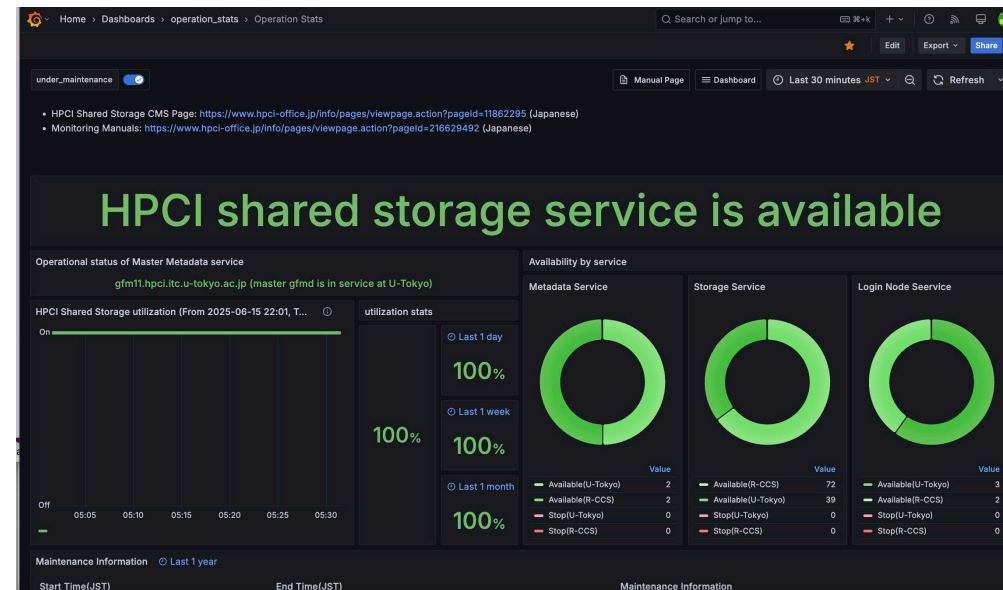
- ユーザ別のダッシュボード環境を自動生成するなど、制限をつけて提供。
- SQLインジェクションが止められない。Influx DBもだけどOSS版の制約がセキュリティ的に弱い。
- TimeScale DBでテーブルにパスワードをかけるなど、セキュリティに注視した環境整備。
- Enterprise検討したが、2020年頃の試算で年1000万と言われたので早急に諦め

LLM連携なども含めて、ダッシュボード環境の変更できないかを本格的に考えている
内部向けのものは問題ないけど、公開系の環境は考えるべき。。

- Fault/Operating Status
- Metadata/File Server Status
- Network
- Storage Capacity
- Capacity Information by User/Group
- Node/SNMP/IPMI/Blackbox Exporters metrics
- Gfarm Exporter(original)
 - Detailed I/O Status and Service Information
- SMART Check Exporter(original)

Future Work:

- (1) Update monitoring using tools such as eBPF, gNMI...
- (2) Introduction of MPC-agent and ML/AI integration



ログ監視環境

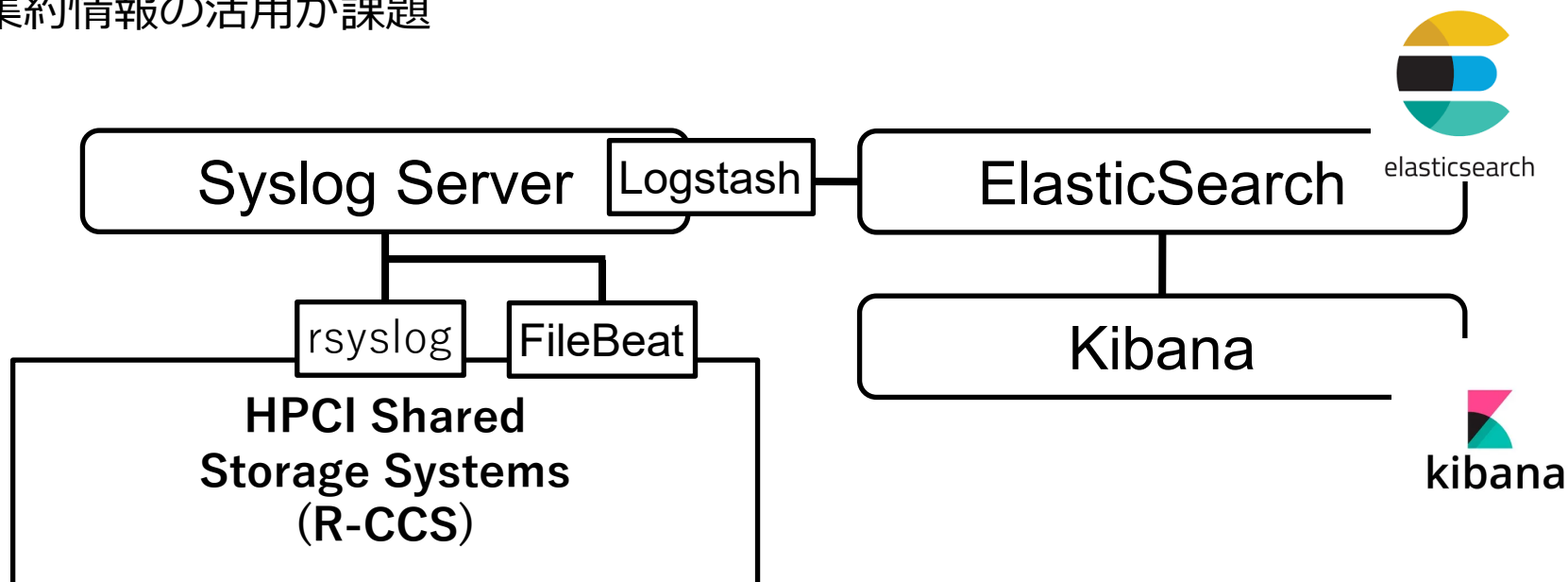
Kibana + ElasticSearchでの監視(基本的なシンプル構造)

HPC環境と違ってサーバ数・仮想環境数は多くない(<100)なのでSyslog Serverでも処理できている。

- Zabbixで対応しているものをこちらに移動させたいと考えている(が、なかなか進んでいない)
- エラー番号毎の集約や、GfarmのI/O情報をログ情報から出力
- SecureログやWeb関連のエラー等の集約に利用

• 課題:

- ログの扱いに制限があり…東西間でのログ共有が進んでいない
→ 一括したログ集約の仕組みが課題となっている。
- 集約情報の活用が課題



Dashboards

Q Search...

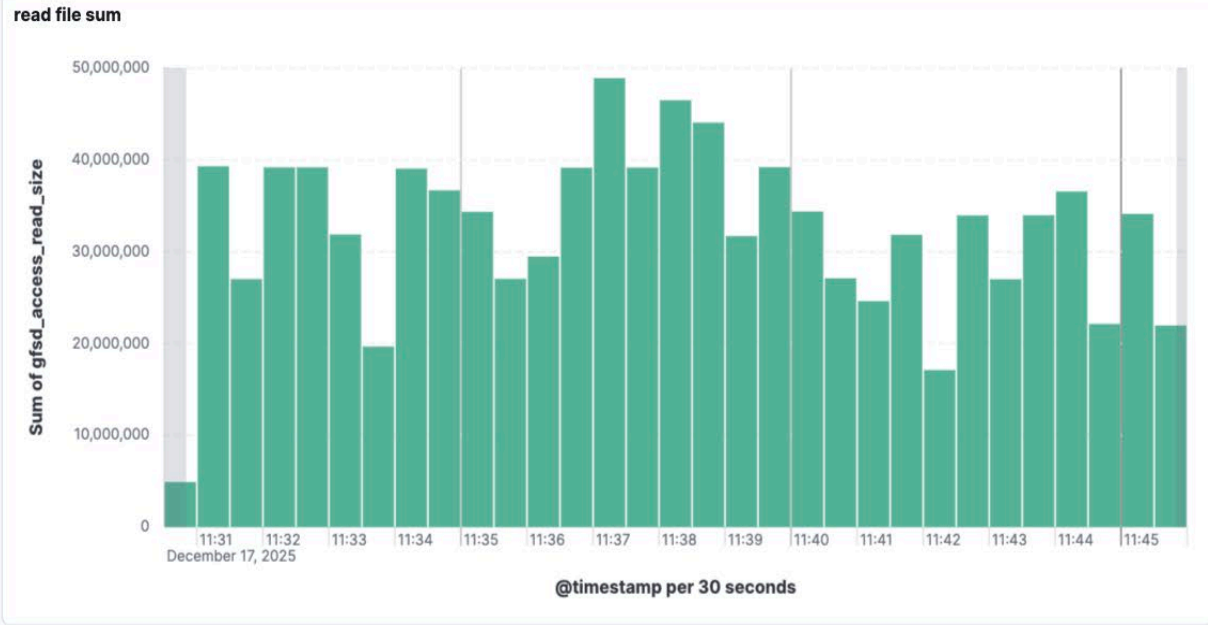
Recently updated Tags Create dashboard

<input type="checkbox"/> Name, description, tags	Last updated ↓	Actions
<input type="checkbox"/> アクセスユーザユニーク数集計グラフ(1時間, 1日, 1週間, 1ヶ月) (4) アクセスユーザ数集計	March 21, 2025	ⓘ
<input type="checkbox"/> アクセスユーザ数集計グラフ(Hourly) (4) アクセスユーザ数集計	March 21, 2025	ⓘ
<input type="checkbox"/> アクセスユーザホスト・ユニーク数集計(1週間、1ヶ月) (4) アクセスユーザ数集計	March 21, 2025	ⓘ
<input type="checkbox"/> replica_check進捗データグラフ(code=1003757) (2) Replica Check向けパーザ	March 24, 2024	ⓘ
<input type="checkbox"/> write_verifyログ集計グラフ (3) Write Verify向けパーザ	March 23, 2024	ⓘ
<input type="checkbox"/> replica_check完了データグラフ(code=1005040) (1) Replica Check向けパーザ(1)	March 23, 2024	ⓘ
<input type="checkbox"/> Read Write Sum	March 29, 2023	ⓘ

-|+

Q Filter your data using KQL syntax

📅 Last 15 minutes Refresh



大きく分けて以下の3点の改善が必須。

- (1) AI・LLM連携**
- (2) 情報収集・集約の強化**
- (3) 提供情報の強化**

特にAI連携は今後必須。

FutureWork: AI/LLM連携

分析レイヤーが足りていないため、分析環境の整備としてLLMの活用を目指す。
一方で、障害等はRAGなどの連携から対応手順の生成やAIエージェント連携等で追加調査の自動化を目指す(運用作業者の負担や、対応レベルを下げる)

(1) 情報分析の強化

- ・ 時系列LLMを用いた将来予測(BigData解析)
 - ・ 容量増加量の予測
 - ・ 特異点(例えば通信速度低下等)の把握
- ・ 障害(特にZabbix系)のアラートへの対応の手順自動作成、自動設定

(2) 可視化ページの自動生成

- ・ メトリクスやログの可視化を自動化

[1] LLM自体による自動化

[2] LLMとダッシュボードソフトウェアを連携させた自動化

[3] 論文や理論、各種オープンソース系の可視化ソフトウェアとの連携を行い適切な情報の提供

→ [1]はOracle DBなどで実現されているもののGUIのソフトウェア埋め込みなどでの連携が課題

[2]は金融系や事務系や、有料のソフトウェアでは実現が進んでいるがオープンソースのダッシュボードなどでは提供されていない傾向 → 特に強化したい箇所

[3]は論文発表のみで一般的なダッシュボードソフトウェア等と連携されていないような分析手法の取り込みをどのように実現するか。



(3) 情報収取の改善

- ・メトリクス等の情報収集をソフトウェアに合わせて改善、エクスポートやパーザー機能の

自動生成の強化

- ・負荷情報を把握しながら、負荷が高い情報の取得や自動試験の実行

(4) 自動設定

- ・設定変更提案・自動設定(アクセス負荷などをみながらgfarmのncpu数や simultaneous replication receiversの値を調整)
- ・ネットワークパラメータの調整
- ・ログインノードの追加やデータのリバランスなどの自動化

時系列LLM

研究対象にはあまりならないかもしれないが、とても強力なツール
Bigデータ解析等に利用。

例: IBM granite

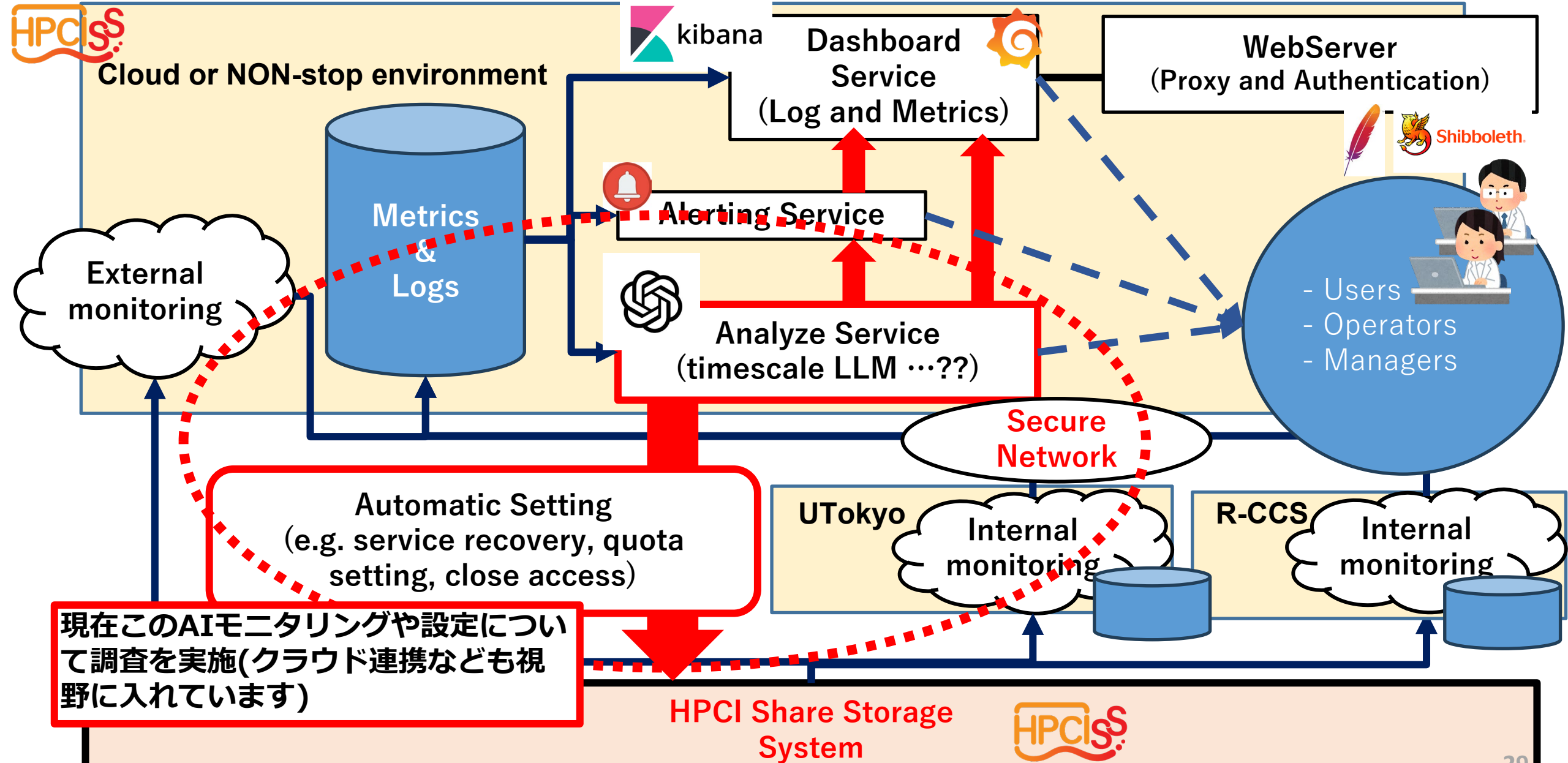
<https://huggingface.co/ibm-granite/granite-timeseries-ttm-r1>

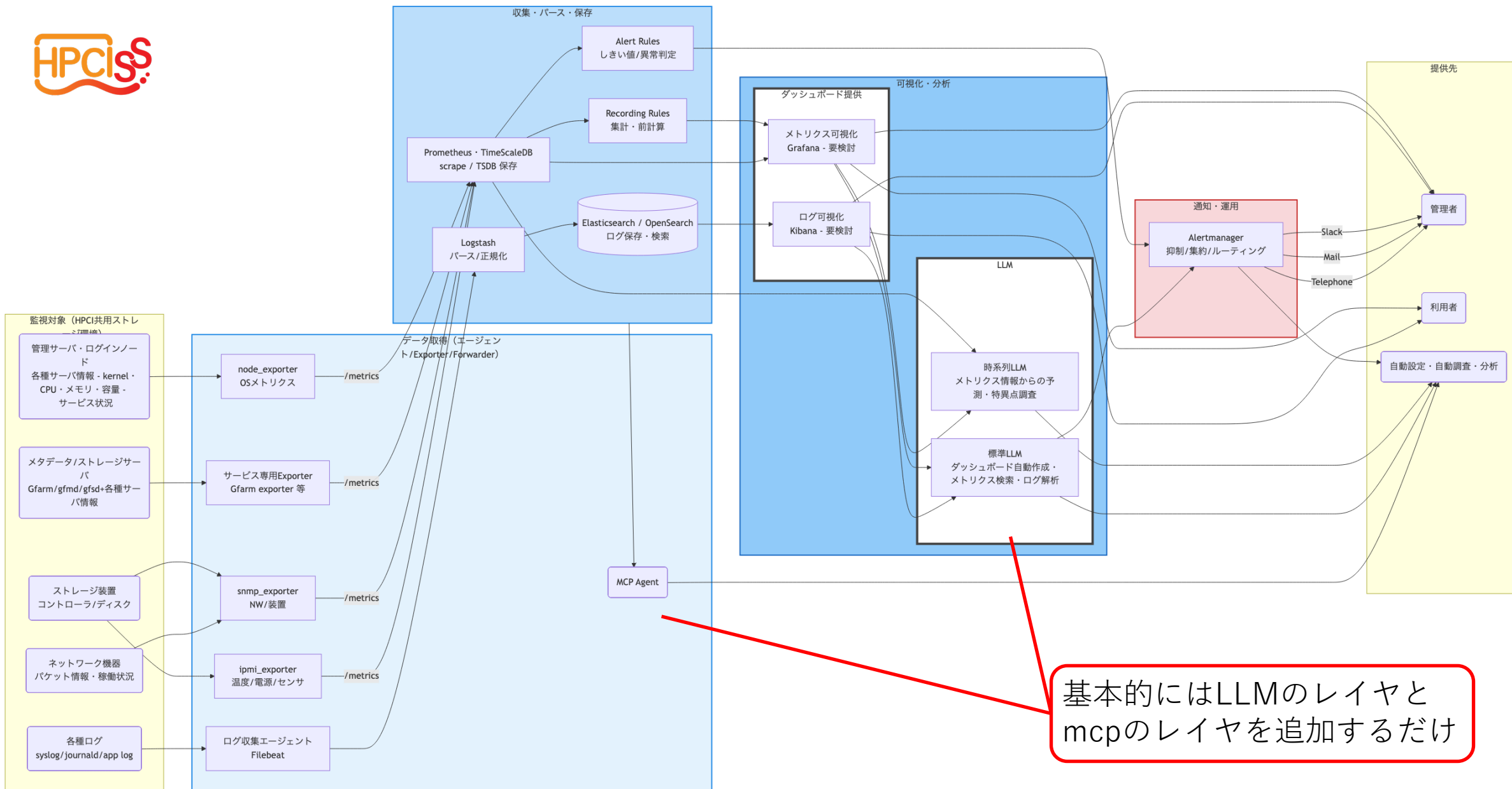
■ 特徴:

- メトリクスデータと時系列データのみでーしたLLMモデル。
- パラメータ数が数百万程度と軽量のものも存在
 - 軽量モデルならCPUでも十分推論できる。
- ZeroShotも強力だがROLAなどの追加学習で予測性能が向上。
(少量チューニングで問題なし)
- 時系列LLMモデルによって、複素系列なども扱い可能

可視化部分については、ダッシュボードや可視化ソフト・ライブラリとの連携が必須。
このため出力された情報、出力する情報をどのように可視化するかは標準LLMモデルなどでプログラム等を自動生成するか、可視化ノウハウに依存。(DBへの投入などの連携が必要)







基本的にはLLMのレイヤと mcpのレイヤを追加するだけ

その他の改善

AIを利用しない改善としては主に以下。

- **データの詳細情報を提供**

- ・ 拡張メタデータの強化とそれに伴うユーザへのメタデータ情報の統計情報等を提供
- ・ (最終的には拡張メタデータ操作での利用を目指したいところ)
- ・ 利用していないデータへの特定や、公開データへのアクセス頻度情報などの提供は必須

- **東西間およびHPCIとのログ・メトリクス情報の共有**

- ・ 現在の規定ではログの提供の制限が厳しい。特に個人情報や利用情報。
- ・ 解析に利用できるようなフォーマットへの変換や、分析情報としての提供を目指したい (オープンデータ)

- **ユーザ情報のAPI化**

- ・ 多くのユーザが複数の環境を利用しているものの、モニタリングが多岐に分かれている。
- ・ API提供などでユーザ利用の情報を、ユーザに簡単に提供する仕組みがあれば…
(例えば富岳のユーザは、富岳のストレージ、ローカルストレージ、HPCI共用ストレージ、
今後はAI4Scienceや量子などのストレージと複数のストレージを活用しないといけないかも)
→ 連携したシステムが理想。
→ モニタリングが別々なのは使いにくい。
(これは我々がベンダーロックインのモニタリングが利用しにくいと思っているのと一緒)

SCA/HPC Asia 2026@Osaka

Date: January 26 ~ 29, 2026

Venue: Osaka International Convention Center

- Theme : Everything with HPC – AI, Cloud, QC and Future Society
- Co-located Events (in progress)
 - **Trillion Parameter Consortium (TPC)**
 - **Accelerated Data Analytics and Computing (ADAC)**
 - **Hpc AlliaNce for Applications and supercoMputing Innovation: the Europe - Japan collaboration (HANAMI)**
 - **ASEAN International HPC School**
 - **R-CCS International Symposium**
- In cooperation with international conferences; ACM, IPSJ, etc.
- Expected number of participants: 1500~3000
- Expected number of exhibits: ~100 companies, universities, and research institutions
- A place where the world's most advanced research and business on supercomputing, AI, big data, cloud, quantum computing, and semiconductors meet!

