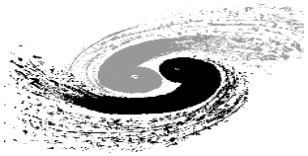
Open maintenance analysis platform at IHEP

Hu, Qingbao (胡庆宝)
huqb@ihep.ac.cn
On behalf of Computing Team of IHEP CC

ISGC 2022

Outline



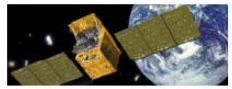
- Computing platform @IHEP
- Challenges & Goals
- Open maintenance analysis platform
- Typical application case
- Summary & Next plan

Computing platform @IHEP

- IHEP is hosting or attending in >15 experiments around the field of high energy physics.
- Distributed Resources
  - Big Data Centers:
    - Beijing IHEP Site
    - Dongguan Data site
  - Exp. Sites:
    - Daocheng(LHAASO), Jiangmen(JUNO) etc.
  - Remote sites:
    - Shandong Univ., Lanzhou Univ. etc
  - Remote sites have good network to IHEP
  - Remote sites is fully managed by IHEP CC
  - Aims to integrate all the sites' resources for Exp.

BESIII (Beijing Spectrometer III at BECPII)

JUNO (Jiangmen Underground Neutrino Observatory)



HXMT(Hard X-Ray Moderate Telescope)

China Spallation Neutron Source

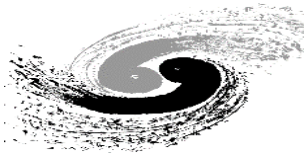
LHAASO (Large High Altitude Air Shower Observatory)

HEPS (High Energy Photon Source)

Monitoring of One Platform for Multiple Data Centers
last 1 week

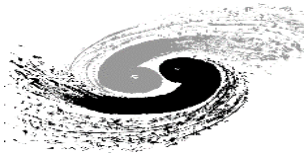
Sites	CPU Resources (CPU Cores)	CPU Resource Utilization	Disk Storage Capacity	Data Storage	Completed Jobs HTC&HPC	Job Run Time (CPU Hour)
IHEPCC	39,996	88.97%	72.12 PB	43.22 PB	2,752,083	5,953,860
DongGuan	31,120	39.51%	6.38 PB	2.30 PB	493	1,399,793
DaoCheng	3,616	47.93%	4.27 PB	3.54 PB	2,036	344,790
CSNS	5,852	31.24%	802.7 TB	375.5 TB	469	290,286
SDU	1,180	8.863%	352.9 TB	238.8 TB	716	16,538
USTC	3,018	26.97%	1.17 PB	767.2 TB	3,539	145,605
LZU	1,768	2.091%	341.8 TB	217.3 TB	585	6,215

Challenges



- More experiments require larger computing platforms
  - Machine scale keeps expanding
  - Cluster environments are more complex
  - The amount of monitoring data continues to increase
  - Monitoring information is more and more closely related
- Traditional monitoring tool
  - Relatively simple functions.
  - Monitoring data cannot be shared with other monitoring tools.
  - Poor scalability for big data scale
- Traditional operation and maintenance technologies cannot guarantee service quality
- More powerful maintenance analysis flatform is needed.

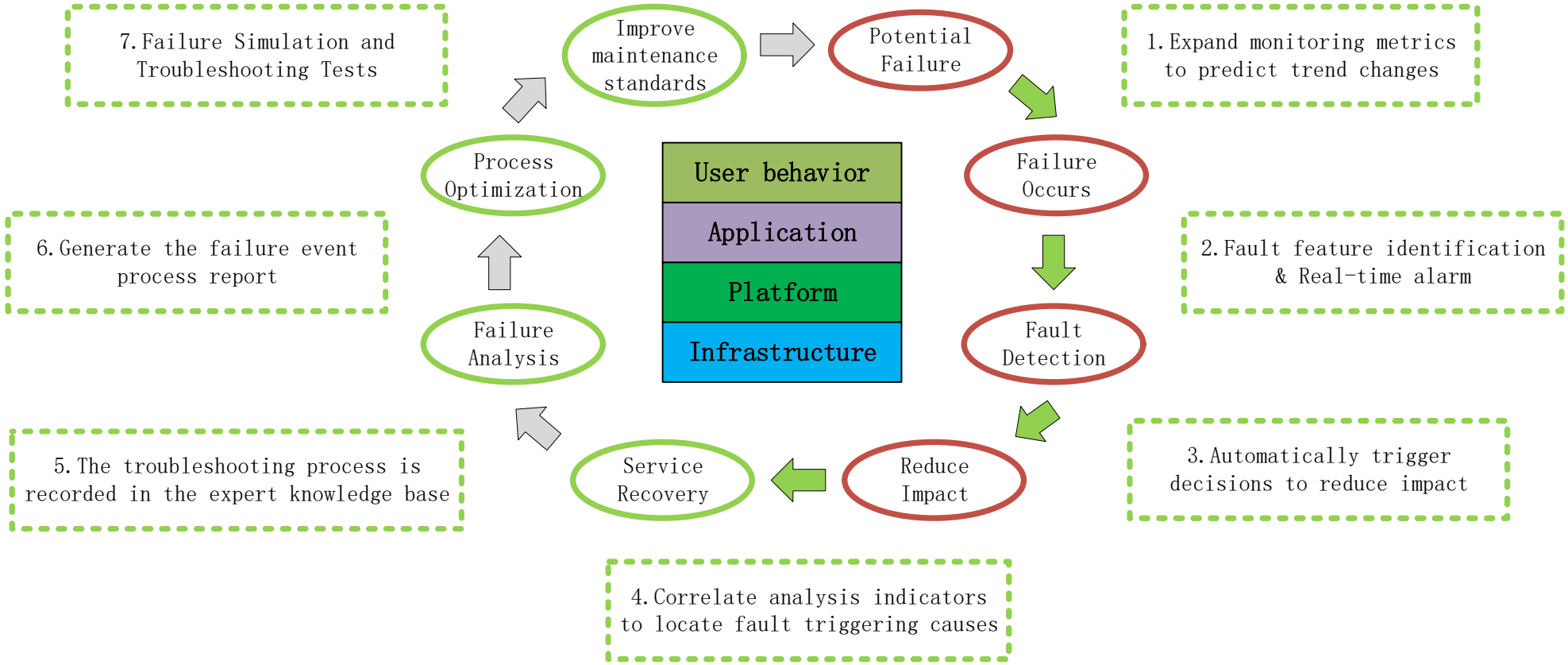
Goals(1/2)



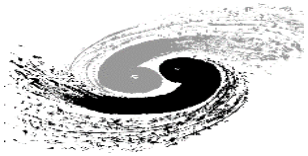
- Basic data management platform
  - Stable provision of "rich and valuable" data
- Intelligent analysis and decision-making platform
  - The value of AI capabilities
  - The accumulated knowledge of operation and maintenance experts
- Agile automation control platform
  - The realization of operation and maintenance capabilities
- Standardized intelligent maintenance management process
  - Realize the closed loop of intelligent maintenance management

Goals(2/2)

- The closed loop of intelligent maintenance management



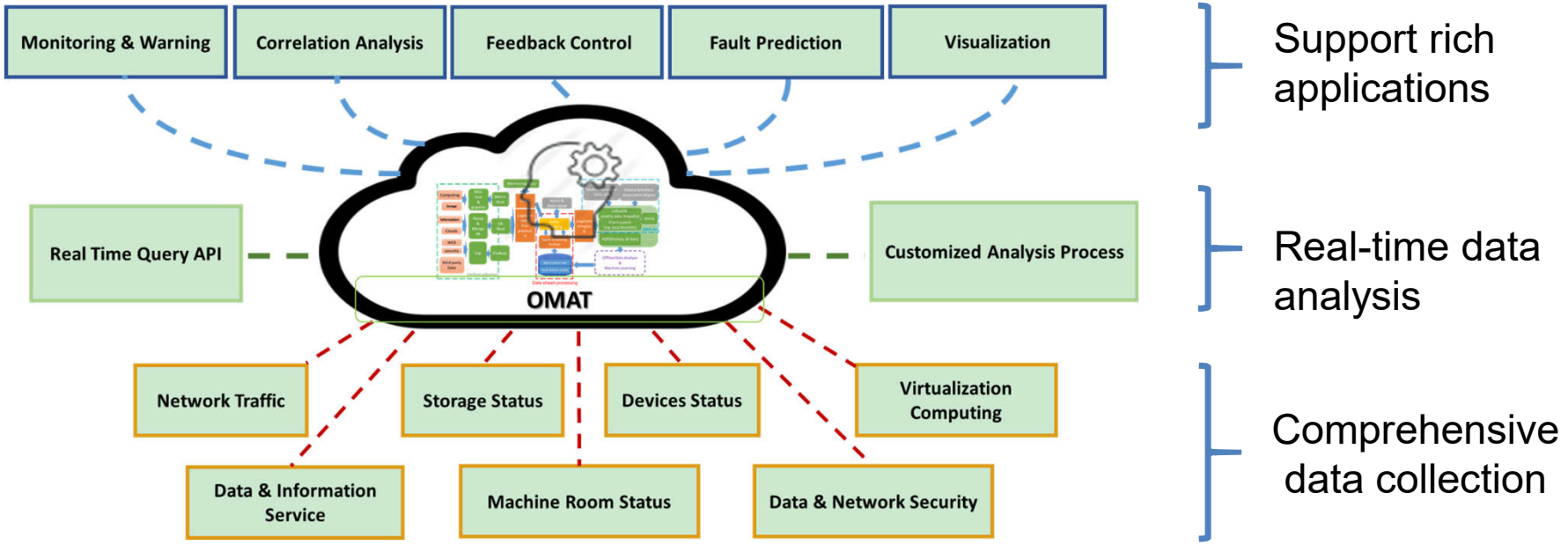
Outline



- Computing platform @IHEP
- Challenges & Goals
- Open maintenance analysis platform
  - Platform Functional
  - Function Realization
  - Platform Performance
- Typical application case
- Summary & Next plan

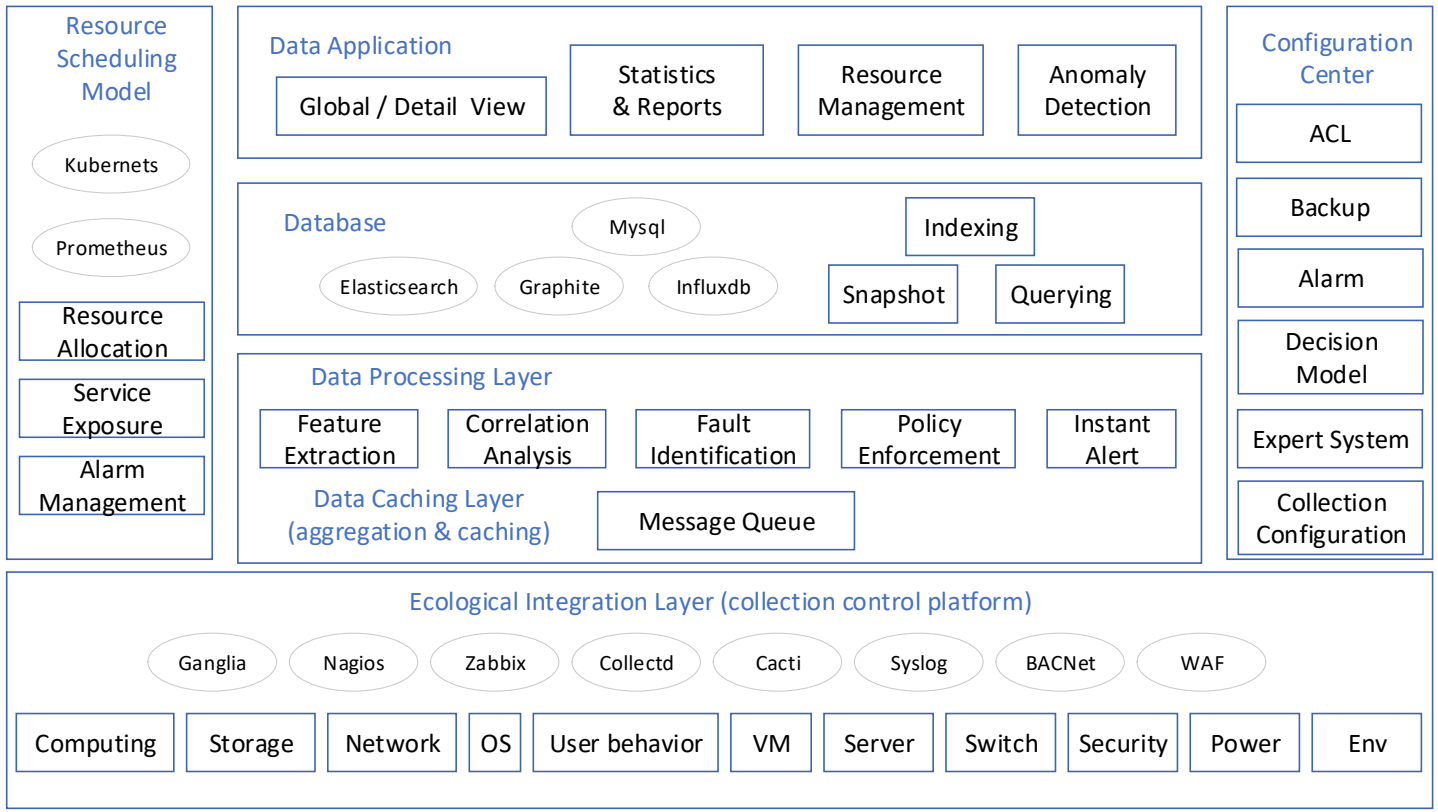
Open maintenance analysis platform

- Maintenance analysis platform is a functional extension of OMAT.
- OMAT (Open Monitoring Analysis Toolkit)
  - Switch “traditional single monitoring” to “whole platform quick analysis”
  - As the core of maintenance analysis platform



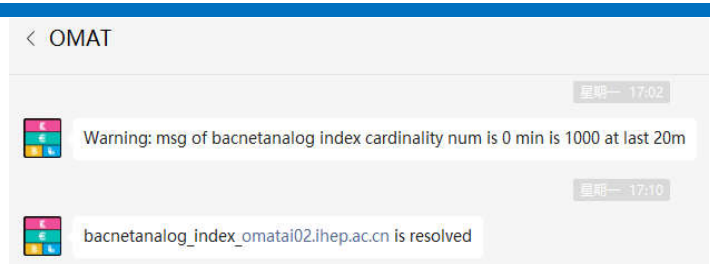
Platform Functional Framework

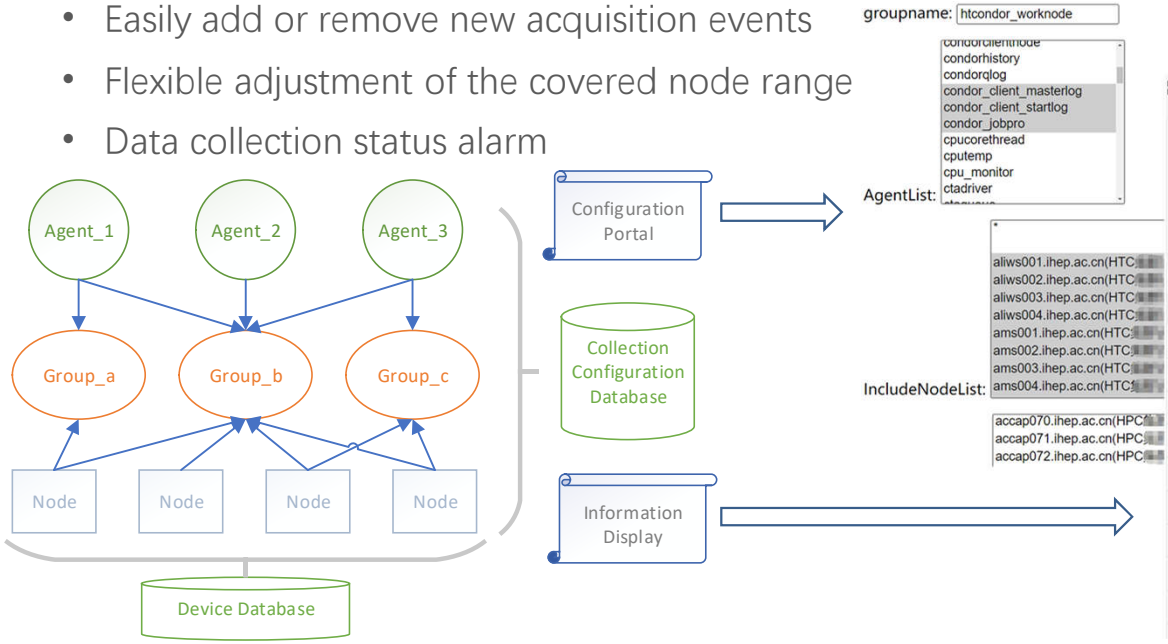
- Platform Features:
  - Comprehensive data collection
  - Flexible and efficient analysis
  - Improve the business value of maintenance data
  - Support rich applications
  - Containerized deployment
  - High reliability and high availability

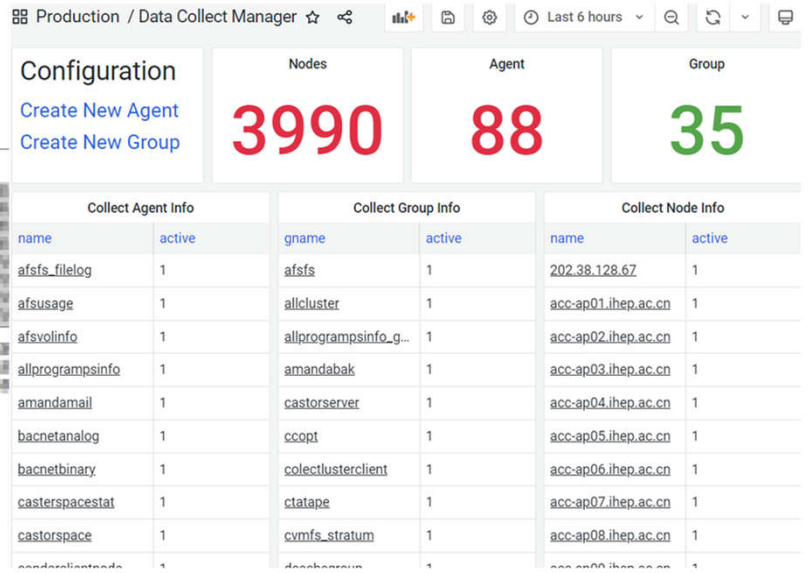


Platform Function Realization (1/4)

- Comprehensive data collection
  - Managed by Collect Control Platform
    - Collect information from nodes and upload to the aggregation module
    - Collected by the traditional monitoring server and forwarded to the aggregation module
    - Easily add or remove new acquisition events
    - Flexible adjustment of the covered node range
    - Data collection status alarm







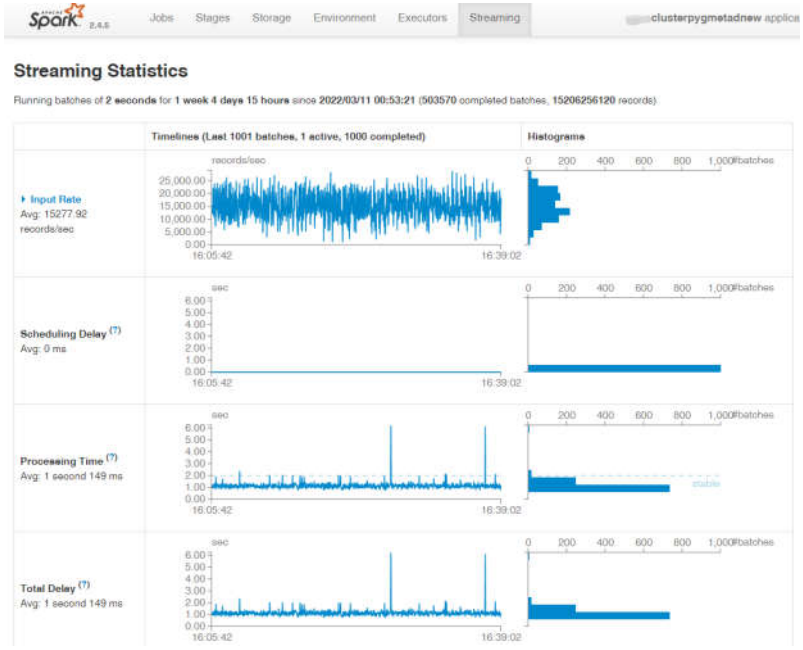
name	active
afsfs_filelog	1
afsusage	1
afsvolinfo	1
allprogrampsinfo	1
amandamail	1
bacnetanalog	1
bacnetbinary	1
casterspacestat	1
castorspace	1

gname	active
afsfs	1
allcluster	1
allprogrampsinfo_g...	1
amandabak	1
castorserver	1
ccopt	1
colectlusterclient	1
ctatape	1
cvmfs_stratum	1

name	active
202.38.128.67	1
acc-ap01.ihep.ac.cn	1
acc-ap02.ihep.ac.cn	1
acc-ap03.ihep.ac.cn	1
acc-ap04.ihep.ac.cn	1
acc-ap05.ihep.ac.cn	1
acc-ap06.ihep.ac.cn	1
acc-ap07.ihep.ac.cn	1
acc-ap08.ihep.ac.cn	1

Platform Function Realization (2/4)

- Flexible and efficient analysis
  - Supported by Apache Spark on k8s with Operator
    - The Spark Operator project was developed by Google and is now an open-source project. It uses Kubernetes Custom Resource for specifying, running and surfacing the status of Spark Applications.
    - Supports collecting and exporting application-level metrics and driver/executor metrics to Prometheus.
    - Configurable restart policy. Scalability, Reliability, Portability
    - Offers us more powerful and stable analysis capabilities and more convenient management
  - Data analysis scripts and dependent analysis environments are deployed on CVMFS. Make sure that the Executor is correctly attached to these paths, and the task can run normally.

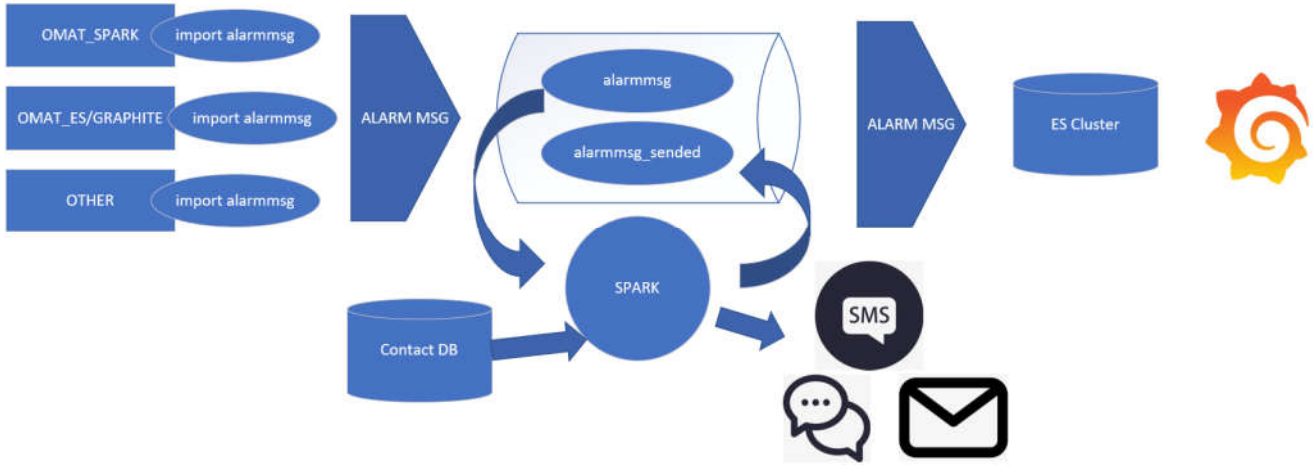



```
(base) [root@omatai01 production]# kubectl get all -n spark-operator
NAME                                      READY   STATUS      RESTARTS   AGE
pod/sparkoperator-7fc547bf98-42hs7        1/1     Running     10         219d
pod/sparkoperator-webhook-init-z52f5      0/1     Completed   0          219d
```

```
(base) [root@omatai01 production]# kubectl get pods -n spark-apps
NAME                                          READY   STATUS    RESTARTS   AGE
pyspark-alarmmsg-1646953451161-exec-1         1/1     Running   0          11d
pyspark-alarmmsg-driver                       1/1     Running   0          11d
pyspark-condorpro-1646441822430-exec-1        1/1     Running   0          17d
pyspark-condorpro-driver                      1/1     Running   0          17d
pyspark-condorqjob-1647923962705-exec-1       1/1     Running   0          12h
pyspark-condorqjob-1647923962705-exec-2       1/1     Running   0          12h
pyspark-condorqjob-driver                     1/1     Running   0          12h
```

Platform Function Realization (3/4)

- Real-time multi-channel message notification
  - Supported by Apache Spark
    - Receive push requests from multiple channels based on kafka.
    - Support rich message push channels such as SMS, Wechat, Email, etc.
    - Support alarm grouping, prevent repeated alarms, etc.
    - Provide fully searchable message history.
    - Convenient alarm contact configuration



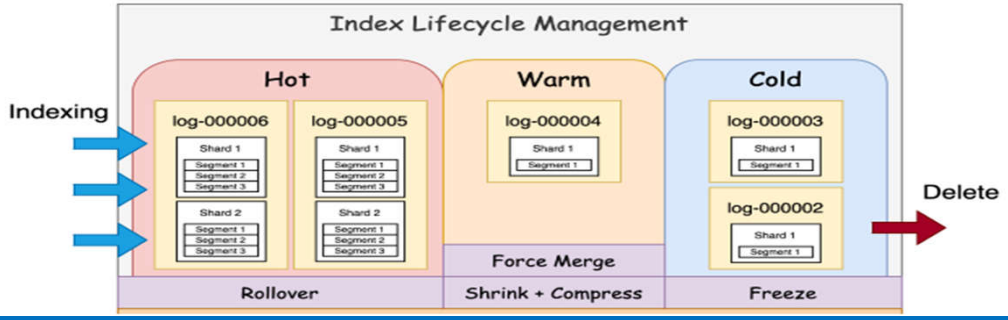
General / OMAT_AlARM ☆ Last 30 days

Alarming Msg List

@timestamp	Content	convergence time	status	name	Contact	source	stat
2022-03-22 14:31:45....	2022-03-22 14:31:45 ...	3 hour	abnormal	lustresyslogalarm_lu...	["wangl","yaoql",...	omatalarm	success
2022-03-22 11:39:26....	2022-03-22 11:39:26 ...	3 hour	abnormal	lustresyslogalarm_lu...	["wangl","yaoql",...	omatalarm	success
2022-03-22 11:39:26....	2022-03-22 11:39:26 ...	3 hour	abnormal	lustresyslogalarm_lu...	["wangl","yaoql",...	omatalarm	success
2022-03-22 11:39:26....	2022-03-22 11:39:26 ...	3 hour	abnormal	lustresyslogalarm_lu...	["wangl","yaoql",...	omatalarm	success
2022-03-22 09:15:01....	indexlist: shrink-ud8_...	1 hour	normal	shrink-ud8_-condorq-...	["huqb"]	omatbackupsnapshot	success
2022-03-22 09:15:01....	indexlist: shrink-ud8_...	1 hour	normal	shrink-igpw-lustreost...	["huqb"]	omatbackupsnapshot	success
2022-03-22 08:45:01....	index: shrink-ud8_-co...	1 hour	abnormal	shrink-ud8_-condorq-...	["huqb"]	omatbackupsnapshot	success

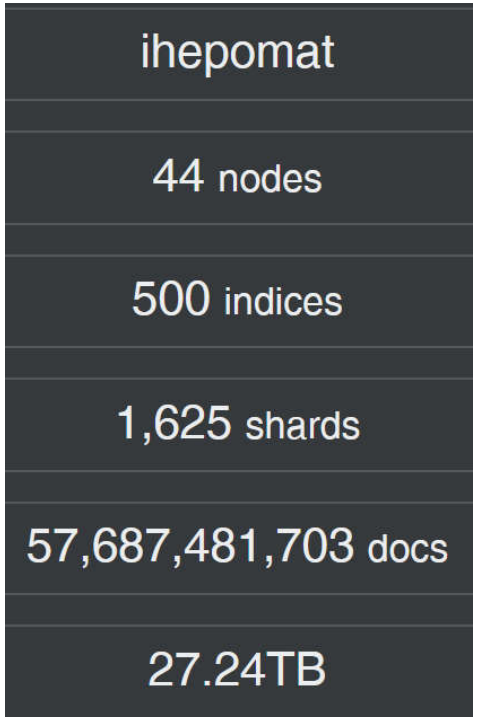
Platform Function Realization (4/4)

- Powerful Data Warehouse
  - Supported by Elasticsearch Cluster and other Database on k8s
    - Efficient data query, divide node roles to handle read and write requests respectively
    - Support data disaster recovery, multiple copies, stored in 3 different node zone
    - Rich data API interface, providing complex query result output based on Django
    - Automatic management of the whole life cycle of index data (with Castor)
      - Forward data flow: Writable hot data (read-write mode + SSD) -> read-only hot data (read-only mode + SSD) -> read-only warm data (read-only mode + mechanical disk) -> read-only cold Data (read-only mode + disk array) -> snapshot backup (does not support query + tape storage)
      - Reverse data flow: snapshot backup (does not support query + tape storage) export -> import database (read-only mode + support query + disk storage) -> provide search service

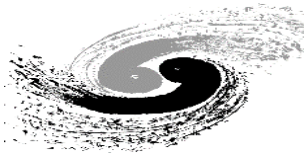


Platform Performance

- Platform performance
  - Maximum data collection capability ~150k doc/s
  - Covering 5k+ nodes in Beijing and remote-site resource sites
  - Average data processing ~60k doc/s Data processing delay ~5s （Spark Streaming on K8s）
  - Maximum data indexing rate ~180k doc/s （44nodes ES cluster on K8s）
  - Covering 400+ types of operation and maintenance data metrics



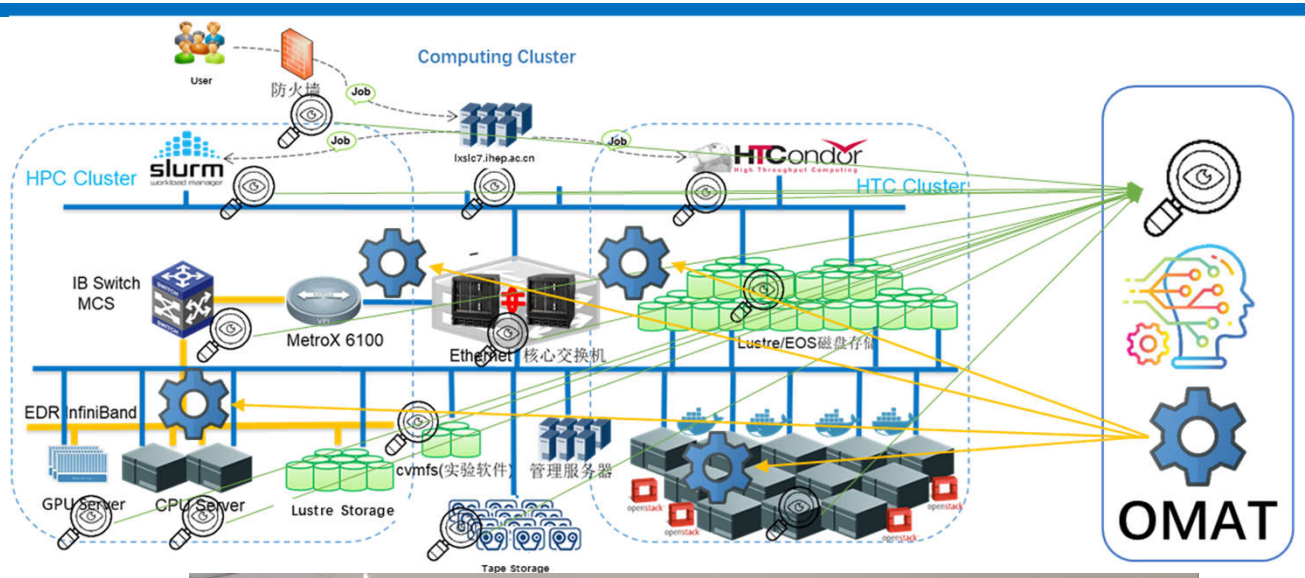
Outline



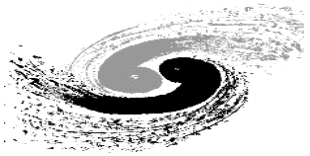
- Computing platform @IHEP
- Challenges & Goals
- Open maintenance analysis platform
- Typical application case
- Summary & Next plan

Comprehensive monitoring of IHEP

- Monitoring field
  - Engine room power environment, PUE
  - Hardware devices, virtual machines, containers
  - System, storage, network performance
  - Job scheduling, job data access behavior, resource management
  - Security authentication, user behavior, network attack
  - Remote site monitoring and other application scenarios



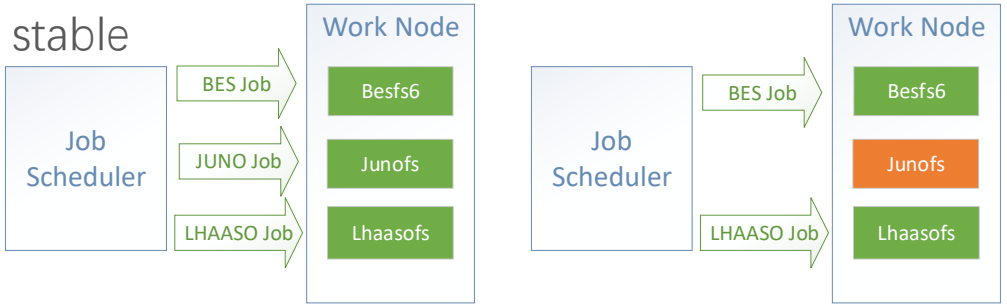
Improve the success rate of user jobs



- Key factors affecting jobs
  - System service exception (Environmental factors)
  - Job program exceptions (Human factors)
- Intelligent maintenance applications are designed to solve these problems
  - Automated cpu resource pool management with quick error detected and responded
  - Anomaly Detection of I/O behaviors based on unsupervised machine learning

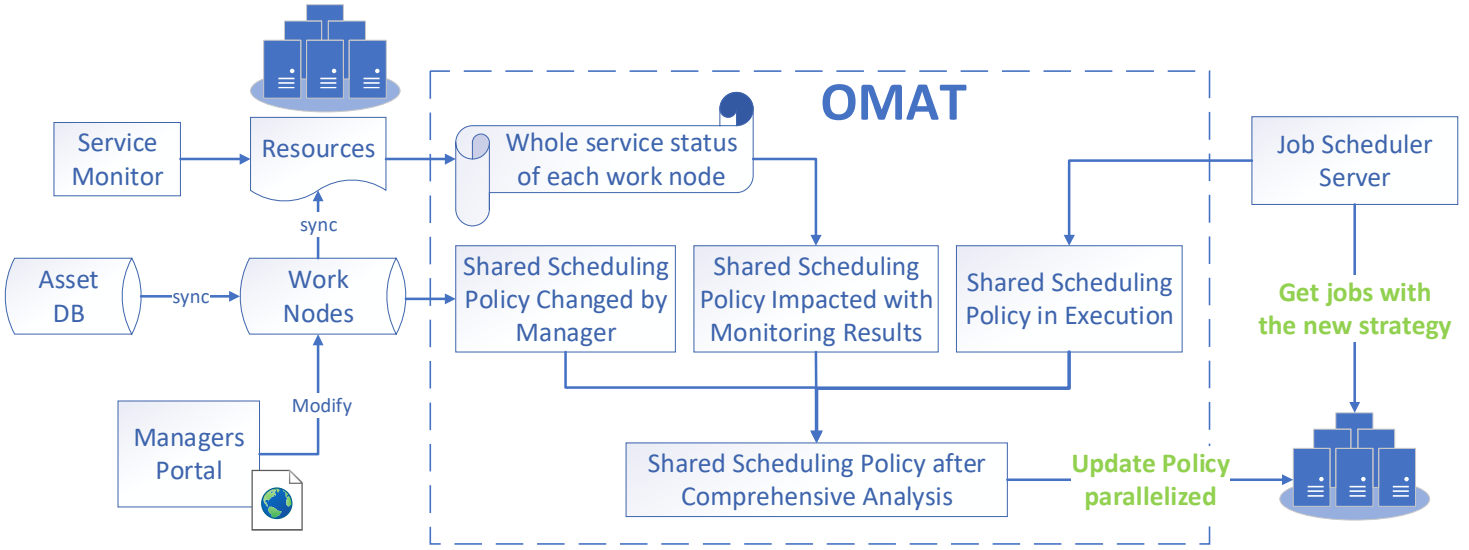
System environmental abnormal (1/2)

- Quick error detected and responded
  - Deal with
    - Unexpected error happened to work node or servers could cause huge jobs failed
    - OMAT detects error in less than 5 sec
    - OMAT inform the job scheduler to evict the error in less than 5 sec
      - Only those job which related to the error will not be dispatched to the error nodes
  - Effect
    - Help the whole computing platform more stable
    - Maximize resource utilization



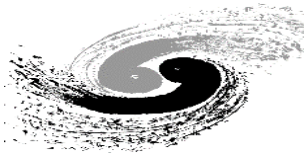
Detail info at https://indico4.twgrid.org/event/14/contributions/360/

System environmental abnormal (2/2)



- Real-time collect whole service status of all work node
  - 50K service status covering 1.5k nodes
- Quickly detect the restored services and new abnormal services.
- Analyze and generate new sharing strategy.
- Based on the high concurrency framework, quickly update multiple node policy files.

User job abnormally (1/2)

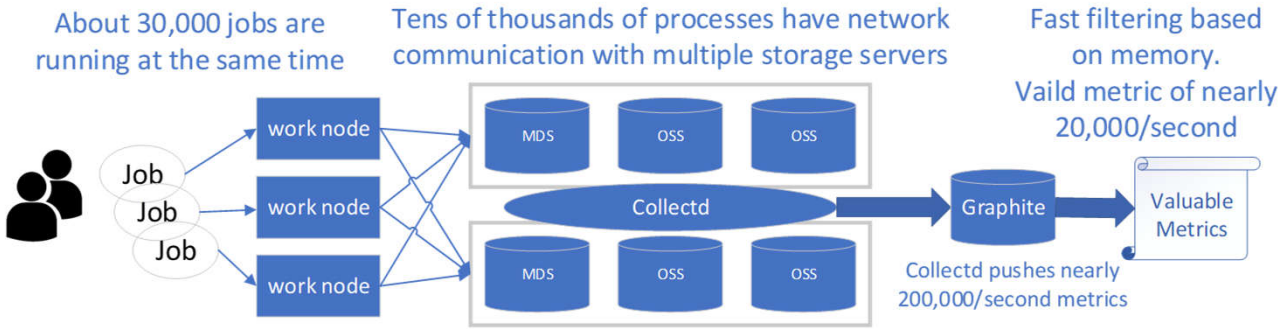


- Problems
  - One or two user abnormal IO behavior could affect the whole file system performance
    - Slow down user normal file access
    - Crash the file servers
  - Traditional way is to analysis storage servers logs, load alarms, and backtracking workflows manually
  - Big MTTR (Mean Time to Restoration), typically in hours
- New solution -- a direct and automatic way of problematic I/O detection
  - Machine learning based on the “ big data collected from the whole platform” by OMAT
  - Reduce MTTR less than 10 mins

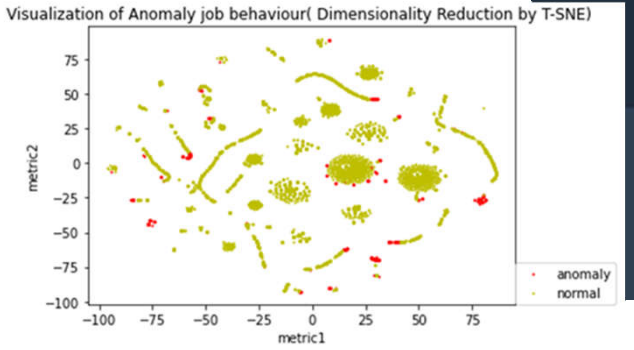
  Detail info at https://indico.cern.ch/event/855454/contributions/4605258/

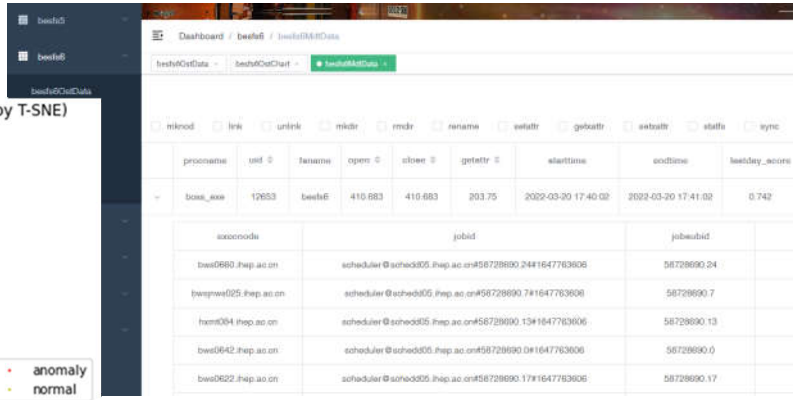
User job abnormally (2/2)

- Collect and analyze the I/O pattern characteristics of the job from Server **by OMAT**
  - 30,000+ jobs in parallel, generating an average of 200,000 file access behavior metrics per second (open, read, seek, write, close…)
- Build proprietary I/O database for each experiment
- Use unsupervised learning algorithms such as isolation forests to find jobs with abnormal I/O behavior.

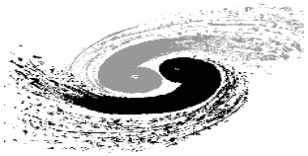



```
202203201752.data:collectd.mds-5_ihep_ac_cn.besfs5-MDT0000-jobstat_starter-suid_12653.derive-statfs,1647769945.000000,0.966666
202203201752.data:collectd.mds-5_ihep_ac_cn.besfs5-MDT0000-jobstat_starter-suid_12653.derive-getattr,1647769945.000000,0.033333
202203201753.data:collectd.mds-6_ihep_ac_cn.besfs6-MDT0000-jobstat_starter-suid_12653.derive-statfs,1647769925.000000,3.183328
202203201753.data:collectd.mds-6_ihep_ac_cn.besfs6-MDT0000-jobstat_starter-suid_12653.derive-getattr,1647769925.000000,0.816665
202203201753.data:collectd.mds-6_ihep_ac_cn.besfs6-MDT0000-jobstat_boss_exe_12653.derive-getattr,1647769985.000000,97.533619
202203201753.data:collectd.mds-6_ihep_ac_cn.besfs6-MDT0000-jobstat_boss_exe_12653.derive-close,1647769985.000000,225.033993
202203201753.data:collectd.mds-6_ihep_ac_cn.besfs6-MDT0000-jobstat_boss_exe_12653.derive-open,1647769985.000000,225.017326
202203201754.data:collectd.mds-6_ihep_ac_cn.besfs6-MDT0000-jobstat_boss_exe_12653.derive-getattr,1647770045.000000,99.082902
```



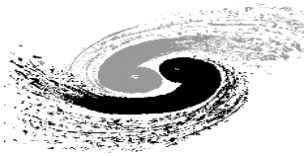


Outline

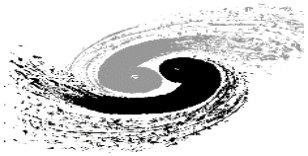


- Computing platform @IHEP
- Challenges & Goals
- Open maintenance analysis platform
- Typical application case
- Summary & Next plan

Summary & Next plan



- Summary
  - Designed and implemented a set of data center intelligent operation and maintenance system, realized the value evolution of massive maintenance data, and solved the increasingly complex maintenance problems of large-scale data centers.
  - The monitoring platform was deployed at IHEP-CC in 2018 and migrated to the container environment at the beginning of last year.
  - Ensure the stable operation of the computing platform.
- Next Plan
  - Build a rich database of abnormal maintenance features, and combine machine learning algorithms to provide intelligent analysis services.



Thanks for your attentions!