

EAST装置技术诊断系统数据高效归档方案设计 设计与实现

侯长申^{1,2}, 奚维斌^{2*}, 杜庆庆², 刘唤唤¹

¹安徽理工大学计算机科学与工程学院, 安徽 淮南

²中国科学院合肥物质科学研究院, 安徽 合肥

收稿日期: 2025年10月1日; 录用日期: 2025年10月28日; 发布日期: 2025年11月5日

摘 要

EAST技术诊断系统需长期监测装置运行状态, 采集的数据类型复杂, 涵盖来自磁体、支撑结构、馈线等多个关键部件的温度、流量、电位等物理信号。每一轮EAST实验运行周期长达5~8个月, 涉及300多个通道、数十种信号源, 采集数据规模可达数10 TB。庞大且持续增长的数据量对系统的归档性能、数据一致性、检索效率以及存储空间管理提出了严峻挑战。研究基于EPICS通过应用Archive Appliance与RDB Archive Engine构建归档系统。实现了对数据的统一管理, 增强了历史数据的多用户并发访问能力与系统整体可扩展性。对Archive Appliance和RDB Archive Engine关于PV最高归档速率、PV检索速度、多客户端同时访问历史数据等方面进行性能测试, 实验结果表明Archive Appliance相较于RDB Archive Engine更适配EAST主机诊断数据的存储需求。

关键词

EAST技术诊断, EPICS, Archive Appliance, RDB Archive Engine

Design and Implementation of an Efficient Data Archiving Solution for the EAST Technical Diagnostic System

Changshen Hou^{1,2}, Weibin Xi^{2*}, Qingqing Du², Huanhuan Liu¹

¹Anhui University of Science and Technology, Huainan Anhui

²Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei Anhui

Received: October 1, 2025; accepted: October 28, 2025; published: November 5, 2025

*通讯作者。

文章引用: 侯长申, 奚维斌, 杜庆庆, 刘唤唤. EAST 装置技术诊断系统数据高效归档方案设计 with 实现[J]. 计算机科学与应用, 2025, 15(11): 60-67. DOI: 10.12677/csa.2025.1511283

Abstract

The EAST technical diagnostic system requires long-term monitoring of the device's operational status. The collected data is complex, encompassing various physical signals such as temperature, flow rate, and electric potential from critical components including the superconducting magnets, support structures, and feeders. Each experimental campaign of EAST lasts from five to eight months, involving over 300 channels and dozens of signal types, with data volumes reaching tens of terabytes. The massive and continuously growing data imposes significant challenges on archiving performance, data consistency, retrieval efficiency, and storage management. This paper presents a data archiving system built on EPICS, utilizing both the Archive Appliance and the RDB Archive Engine. The system achieves unified data management and improves historical data access concurrency and overall scalability. Performance tests were conducted focusing on maximum PV archiving rate, PV retrieval speed, and multi-client access to historical data. The results demonstrate that the Archive Appliance is better suited for the storage requirements of EAST's host diagnostic data compared to the RDB Archive Engine.

Keywords

EAST Technical Diagnostics System, EPICS, Archive Appliance, RDB Archive Engine

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

EAST (Experimental Advanced Superconducting Tokamak)是中国自主设计、建造, 并成功投入实验运行的世界首台全超导托卡马克[1]。技术诊断系统负责采集超导线圈、支撑结构、馈线等关键部件的温度、电阻、电压等数据, 为装置运行安全提供支撑[2]。随着监测通道数量和采样频率不断增加, 传统归档系统在数据一致性、检索效率及可扩展性方面已难以满足需求, 亟需构建高效、稳定的归档方案。针对大规模实验装置产生的时序数据, 业界已有多种开源时序数据库(TSDB)方案, 如 InfluxDB、Prometheus、TimescaleDB 等。这些系统在工业监控领域性能优异, 但与 EPICS 控制框架兼容性较差。考虑到 EAST 技术诊断系统基于 EPICS 架构, 本文选取其生态中的两种典型归档组件——Archive Appliance (AA)与 RDB Archive Engine (AE)进行对比研究。二者均能直接对接 IOC 层 PV 数据流, 具有代表性与工程实用性。通过 softIOC 生成模拟诊断数据, 分别基于 AA 和 AE 实现归档, 并利用 Phoebus 客户端对其历史数据查询性能与系统资源占用进行评估, 以验证两种方案在 EAST 主机诊断数据归档中的适用性。

2. EPICS 架构

EPICS 全称为“Experimental Physics and Industrial Control System”, 是上世纪 90 年代初由美国洛斯阿拉莫斯国家实验室(LANL)和阿贡国家实验室(ANL)等联合开发的大型控制软件系统, 许多大型科学研究工程项目都开始采用, 比如中科院高能物理研究所的加速器控制系统, 同步辐射实验室的同步辐射光源装置等[3] [4]。

EPICS 软件结构分为四层: 驱动层负责与底层硬件设备进行交互, 该层将各类控制设备(如 PLC、AD/DA、采集卡等)统一封装为标准接口, 向上提供数据读写功能。数据处理层是 EPICS 的核心逻辑层,

通过各种记录类型(如 ai、ao、bi、bo、calc 等)定义 PV 的处理方式和属性。IOC 层是 EPICS 系统的执行核心,负责加载数据库文件、初始化驱动模块并运行数据处理逻辑[5] [6]。每个 IOC 运行一个独立的控制进程,管理本地 PV 并与外部系统通信。通信与客户端层作为最上层负责实现 IOC 与客户端之间的数据通信和交互,主要基于 Channel Access 协议,是系统的人机交互接口。关于 IOC 软件结构见图 1。

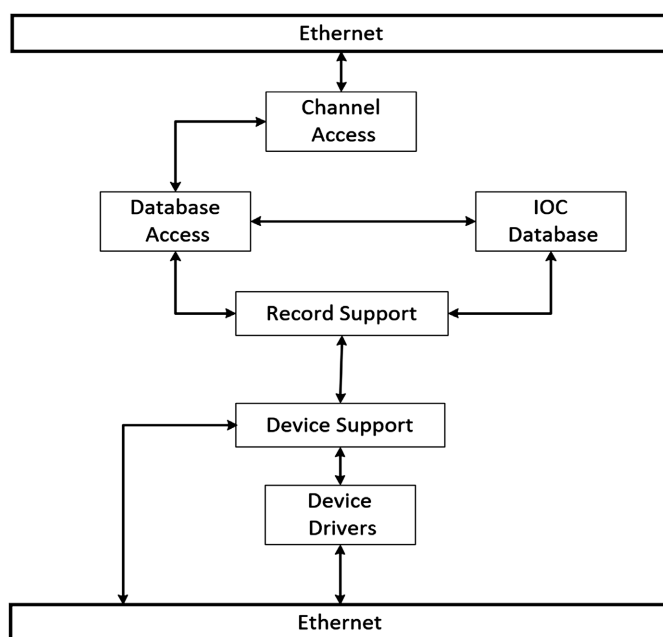


Figure 1. IOC software architecture

图 1. IOC 软件结构

3. EAST 技术诊断数据的需求分析

(1) EAST 技术诊断数据如果直接存储为原始格式(例如二进制流、PLC 内部寄存器值),后续的归档、查询、显示都会因为不同设备、不同系统之间的存储数据格式和访问方式差异很大导致复杂化。因此,需要将底层的多种 EAST 诊断数据类型转化为一种统一的数据类型进行存储,方便后续对数据的管理。

(2) 连续变化类信号(如温度、电阻)通常变化平缓,需要完整记录其变化趋势,用于分析热负荷、材料应力等,突变类信号(如超导线圈失超探测、电流快速上升)可能在极短时间内发生剧烈变化,若采用固定周期采样,可能错过关键瞬间。因此,需要采用基于变化采样和周期采样策略。

(3) 数据存储采用传统关系型数据库直接存储技术诊断数据,不仅磁盘空间消耗巨大,而且容易因数据表过大而导致写入与访问性能显著下降。在实验分析过程中,科研人员通常需要根据数据变量名称与时间范围快速检索历史数据。若数据集中存放在单一数据库表或少量文件中,查询往往需要长时间的扫描,就会导致数据检索速度慢,系统响应时间长[5]。因此,在数据存储方面需要采用数据压缩和分区存储策略。

(4) 多用户同时访问诊断数据监控界面时,系统响应速度明显下降,部分用户可能出现数据无法加载或超时的情况。此外,现有的数据展示方式分散、界面功能单一,缺乏统一的可视化管理平台。因此,系统迫切需要建立一个统一的数据展示平台,以满足多用户并发访问的需求。

4. EAST 技术诊断数据归档方案的设计与实现

在设计 EAST 技术诊断数据存储方案时,需要全面梳理数据从产生、转换、传输直至存储的完整流程,并结合数据流向图如图 2 明确诊断系统中以过程变量(PV)为核心的存储需求。同时,应对 EAST 技

术诊断数据的特性进行深入分析,并针对当前数据存储中存在的问题提出相应的解决策略。鉴于 EAST 主机诊断系统尚未全面开放实时数据接口,因此在实验阶段采用 softIOC 生成模拟 PV 信号进行验证。模拟信号设计尽量贴近真实诊断环境:包括缓慢变化的温度信号、周期性波动的流量信号以及突变型的电压失超信号等,以覆盖典型的运行工况特征。通过这种方式,可以在不依赖真实硬件的前提下,对归档系统的采样稳定性、数据完整性及查询性能进行定量测试。这种基于模拟环境的方案既降低了实验风险,又为后续接入 EAST 实机信号提供了验证基础与参数参考。基于 AA 和 AE 的整体设计如图 3 和图 4。



Figure 2. EAST technology diagnostic data flow diagram

图 2. EAST 技术诊断数据流向图

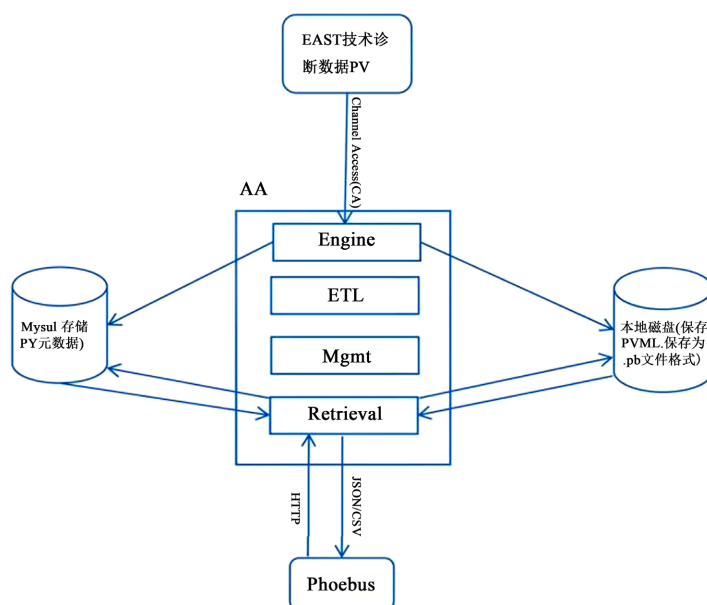


Figure 3. AA archiving overall logic diagram

图 3. AA 归档整体逻辑图

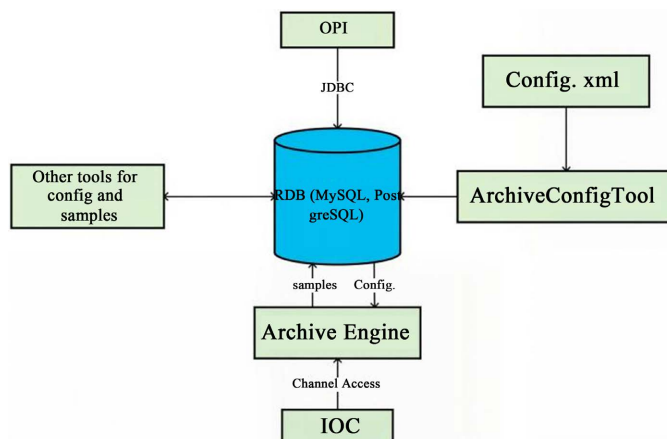


Figure 4. Archive Engine archiving principle diagram

图 4. Archive Engine 归档原理图

4.1. 基于 Archiver Appliance (AA)归档方案的设计与实现

本方案采用 EPICS 社区广泛使用的 Archiver Appliance (AA)作为核心归档引擎[7] [8]。AA 采用 B/S 架构, 通过将 WAR 包部署于 Tomcat 服务器即可提供完整的归档服务。编译 Archiver Appliance 源码包将其部署在 Tomcat 上, 执行脚本 single_machine_install.sh 利用 mysql-connector-java.jar 将 AA 与 MySQL 连接, 配置环境变量并设置短、中、长期存储位置 export ARCHAPPL_SHORT_TERM_FOLDER=/usr/local/Archiver/store/STS、export ARCHAPPL_MEDIUM_TERM_FOLDER=/usr/local/Archiver/store/MTS、export ARCHAPPL_LONG_TERM_FOLDER=/usr/local/Archiver/store/LTS, 运行修改好的 sampleStartup.sh 脚本, ./sampleStartup.sh start/stop/restart, 启动 Archiver Appliance。



Figure 5. Main interface of the AA management platform
图 5. AA 管理平台主页

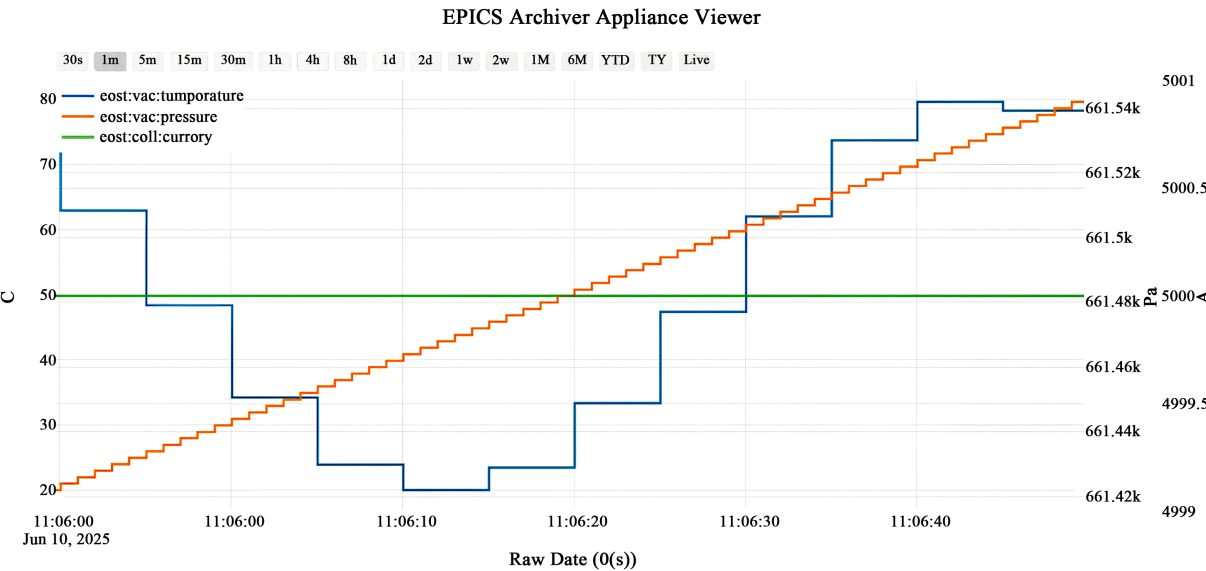


Figure 6. Comparison of historical trends of multiple PVs
图 6. 多 PV 对比历史曲线图

AA 采用三级数据分层存储策略, 分别为 shortTerm、mediumTerm 和 longTerm。Engine 模块将采集到的 PV 数据以.pb 文件格式写入 shortTerm 路径, 由 ETL 模块根据时间策略定期迁移至 mediumTerm 和 longTerm, 实现数据的高效管理与按需读取, 提升了系统的响应效率与存储可扩展性。打开浏览器访问 <http://localhost:17665/mgmt/ui/index.html> (如果 AA 不在本机需要将 localhost 替换为 AA 所在主机的 IP) 获取 AA 的归档和查询界面, 如图 5 和图 6 所示, 在界面中进行诊断数据 PV 的归档、采样策略、历史数据检索等操作。

在 AA 管理平台中, 用户可以对已归档的 PV 执行多种管理操作[9][10]。“修改归档参数”功能, 可以调整 PV 的采样方式(如 Monitor 或 Scan)、采样周期以及是否由其他 PV 控制归档条件; “暂停归档”用于临时停止某个 PV 的数据采集与存储, 适用于调试或暂时无需记录数据的场景; “恢复归档”功能, 使被暂停的 PV 再次开始归档; “删除 PV”功能可以将其从系统中移除, 虽然 PV 被移除, 但其历史归档文件仍会保存在存储路径中; “数据合并”用于将 PV 的多个历史.pb 数据文件进行整合, 以提升检索性能; 此外, 系统还支持“重命名 PV”, 可在不影响数据的基础上对 PV 名称进行逻辑调整, 方便后续统一管理与识别。

为实现历史数据的可视化检索与趋势展示, 需在 Phoebus 客户端中配置 Retrieval 接口地址。在 setting.ini 文件里配置 URL: pbraw://localhost:17668/retrieval 用于指示客户端通过 PB RAW 协议从本机运行的 Archiver Appliance 中提取历史数据, 是历史查询功能正常工作的前提条件之一。Phoebus 与 Archiver Appliance 成功连接, 如图 7。

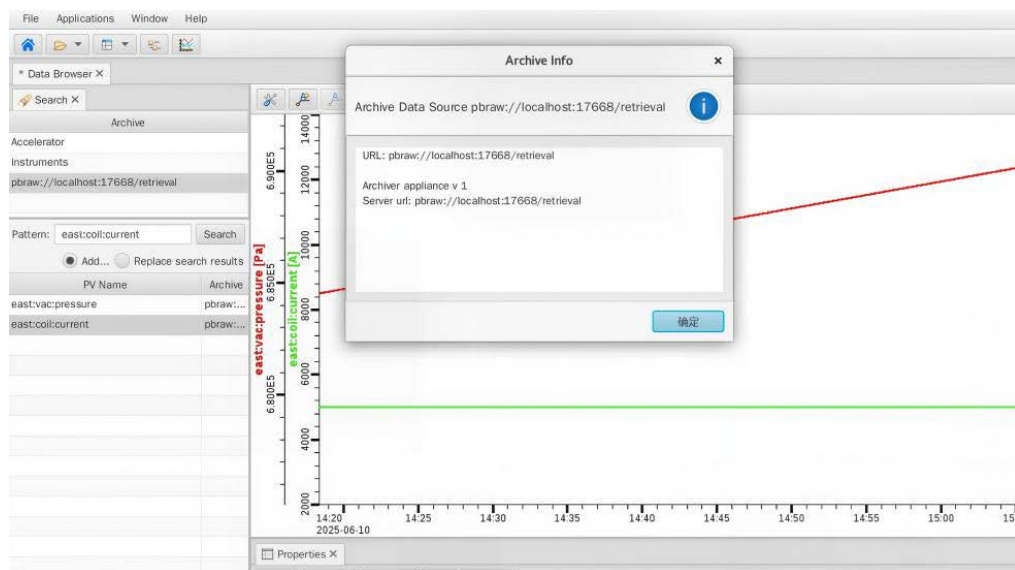


Figure 7. Phoebus retrieval interface based on the Archiver Appliance

图 7. 基于 Archiver Appliance 的 Phoebus 检索图

4.2. 基于 Phoebus Archive Engine (AE)归档方案的设计与实现

(1) 在 MySQL 里创建 archive 数据库和密码为 \$archive 的 archive 用户, 在 archive 数据库里创建归档 PV 所用的表, 参考 Phoebus 里的 MySQL.dbd 文件[11]。

./archive-engine.sh -list 列出数据库中已注册的归档引擎实例

./archive-engine.sh -engine Demo -export Demo.xml 导出 Demo 实例配置文件 Demo.xml 进行修改

./archive-engine.sh -engine Demo -import Demo.xml -port 4812 -replace_engine 重新导入 Demo.xml 并替代原始文件(设置引擎监听的端口号为 4812)

`./archive-engine.sh -engine Demo-port 4812` 启动归档引擎加载 Demo 实例，如果不指定.ini 数据库配置文件，就默认加载内置的.ini 配置文件，通过 archive 用户名和密码与 archive 数据库连接，内置的.ini 文件连接数据库配置为：

```
org.csstudio.trends.databrowser3/urls=jdbc:mysql://localhost/archive|RDB*xnds://localhost/archive/cgi/ArchiveDataServer.cgi
```

```
org.csstudio.trends.databrowser3/archives=jdbc:mysql://localhost/archive|RDB*xnds://localhost/archive/cgi/ArchiveDataServer.cgi
```

`./phoebus.sh` 启动 Phoebus，如果不用-setting 参数指定加载.ini 文件，也会默认加载内置.ini 文件使得 Phoebus 客户端与 MySQL 连接。

(2) 在 PostgreSQL 里创建 tsarch 数据库、密码为\$archive 的 archive 用户、密码为\$tsarch 的 tsarch 用户，在 tsarch 数据库里创建归档 PV 所用的表，参考 Phoebus 里的 setup.sql 文件[12]。(Archive Engine 用 tsarch 用户连接 tsarch 数据库，Phoebus 用 archive 用户连接 tsarch 数据库)授予用户 archive 当前数据库中 public schema 下所有表的所有权限。

```
psql -U postgres
```

```
CREATE USER archive WITH PASSWORD '$archive';
```

```
\q
```

```
psql -U postgres -d tsarch
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO archive;
```

使用-settings 参数将 Archive Engine 与 tsarch 数据库连接并向其导入 Demo.xml，然后重新运行 Archive Engine 加载 Demo 实例，则成功实现归档。使用 `python3 create_settings_template.py` 命令编译此.py 脚本文件生成 settings_template.ini 文件，在.ini 文件添加 Phoebus 连接 tsarch 数据库的语句：

```
org.csstudio.trends.databrowser3/urls=jdbc: postgresql: //localhost: 5432/tsarch|RDB
```

```
org.csstudio.trends.databrowser3/archives=jdbc: postgresql: //localhost: 5432/tsarch|RDB
```

`./phoebus.sh -settings settings_template.ini` 启动 Phoebus 并加载.ini 文件后成功与 PostgreSQL 里的 tsarch 数据库连接，如图 8。

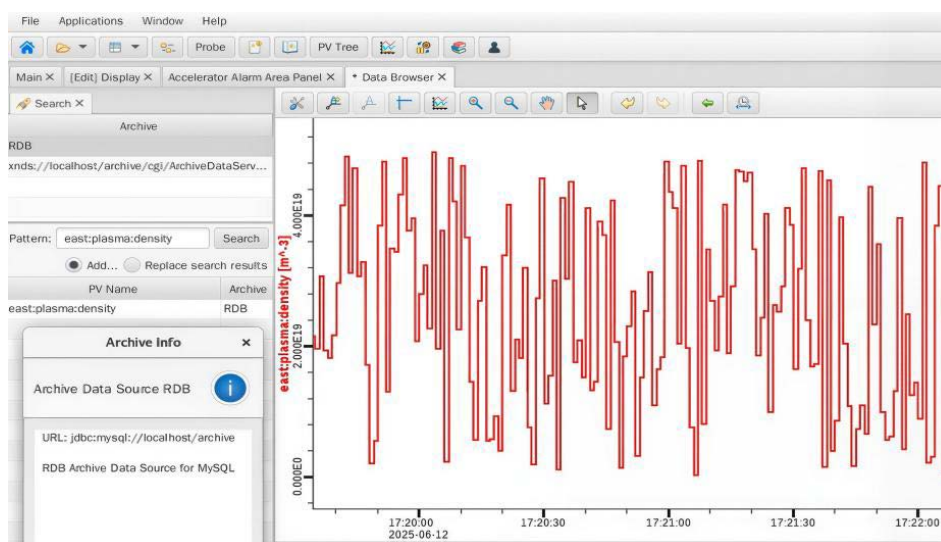


Figure 8. Phoebus retrieval interface based on the Archive Engine

图 8. 基于 Archive Engine 的 Phoebus 检索图

5. 两种归档方案性能对比分析

针对 EAST 主机诊断数据特点并结合表 1 中实验结果来看，采用基于 Archive Appliance 的归档方案相较于采用 RDB Archive Engine 的归档方案更能满足 EAST 技术诊断数据的存储需求。

Table 1. Experimental results and comparison
表 1. 实验结果及对比

对比维度	Archive Appliance (AA)	Archive Engine (AE)
单个 PV 最大采样率	1000 Hz	<1000 Hz
最大可归档 PV 数量	上万级别	上千级别
查询方式	高效的 HTTP REST 接口	依赖 SQL 查询
查询性能(1 Hz, 1 天数据)	<0.5 秒	1~2 秒
系统架构	模块化、多线程、分布式，可灵活调配 CPU、内存和磁盘资源	架构简单，缺乏扩展性，存在资源瓶颈
适用场景	大规模实验装置，需高频采样、大规模并发归档与查询	小型实验室或单点部署，归档规模有限

6. 总结

本文实现了一个具备软 IOC 模拟、双归档方案写入、Phoebus 客户端展示的完整实验框架，为 EAST 主机诊断数据归档系统的实际部署提供了基础。未来将尝试接入真实硬件信号，进一步验证系统稳定性与数据一致性。

参考文献

[1] 万宝年, 徐国盛. EAST 全超导托卡马克高约束稳态运行实验研究进展[J]. 中国科学: 物理学力学天文学, 2019, 49(4): 47-59.

[2] 陈灼民, 钱静, 龙凤, 等. 全超导托卡马克装置(EAST)的技术诊断系统[J]. 低温与超导, 2007(2): 93-95, 102.

[3] 彭杨, 肖炳甲, 王华忠, 等. 基于 EPICS 框架的 EAST 数据服务器监控系统实现[J]. 微计算机信息, 2011, 27(10): 25-27.

[4] 李丹清, 韩利峰, 陈志军, 等. 基于 EPICS 高温制氢控制系统的设计与实现[J]. 仪器仪表用户, 2019, 26(11): 10-15, 98.

[5] 曾飘飘. 基于 EPICS 的 HIRFL 真空监测系统设计与实现[D]: [硕士学位论文]. 北京: 中国科学院大学(中国科学院近代物理研究所), 2023.

[6] 刘聪. 基于 EPICS 的红外测温与高速数据存储系统应用研究[D]: [硕士学位论文]. 兰州: 西北师范大学, 2020.

[7] EPICS Archiver Appliance (2024) Official Documentation. <https://github.com/slacmshankar/epicsarchiverap>

[8] 赵子龙, 徐慧超, 龚培荣. 基于 Archiver Appliance 的束线数据管理系统[J]. 核技术, 2018, 41(3): 11-16.

[9] Schmidt, V. (1998) The Development of the JET Control and Data Acquisition System. *IEEE Transactions on Nuclear Science*, 45, 2026-2032. <https://doi.org/10.1109/23.710984>

[10] Steven, S.W.R. (2022) TCP/IP Illustrated Volume 1: the Protocols. Addison Wesley.

[11] Phoebus RDB Archive Engine (2024) CS-Studio Documentation. <https://control-system-studio.readthedocs.io/en/latest/>

[12] 郭冰, 魏永波. 基于 PostgreSQL 的 CRDM 数据存档系统[J]. 核技术, 2018, 41(2): 67-70.