



Systems for Monitoring and Analysing the Synchronization Network in Critical Facilities

Eusebio N. Al-Soliman, Miguel Damas, Carlos Megías and Javier Díaz

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 6, 2024

Sistemas para la monitorización y análisis de la red de sincronización en infraestructuras críticas

Eusebio N. Al-Soliman Fuentes
Ingeniería de Computadores, Automática y Robótica
Universidad de Granada (UGR)
Granada, España
eunaif@ugr.es

Carlos Megías Núñez
Ingeniería de Computadores, Automática y Robótica
Universidad de Granada (UGR)
Granada, España
narg@ugr.es

Miguel Damas Hermoso
Ingeniería de Computadores, Automática y Robótica
Universidad de Granada (UGR)
Granada, España
mdamas@ugr.es

Javier Díaz Alonso
Ingeniería de Computadores, Automática y Robótica
Universidad de Granada (UGR)
Granada, España
jda@ugr.es

Resumen—Dentro del ámbito de las instalaciones científicas, hay sistemas críticos que requieren de un control robusto para su correcto funcionamiento. Tal es el caso de las redes de sincronización, que permiten tener todos los dispositivos de la infraestructura sincronizados en tiempo para poder realizar mediciones de calidad. Para estas redes de sincronización entran en juego principalmente dos protocolos, NTP (*Network Time Protocol*) y PTP (*Precision Time Protocol*), que deberán ser monitorizados y controlados junto al resto de sistemas de la instalación. En la industria esto se realiza con los software SCADA (*Supervisory Control And Data Acquisition*). En el transcurso de este artículo, se va a realizar una pequeña simulación de una red de sincronización de dispositivos que será monitorizada con dos software SCADA: EPICS (*Experimental Physics and Industrial Control Systems*) y WinCC Open Architecture. Se ilustrará el proceso seguido para poder monitorizar la red de sincronización en sendos SCADAs, destacando las principales características de cada uno y obteniendo unas conclusiones finales sobre qué sistema es el más conveniente para el caso de estudio propuesto.

Palabras clave—red de sincronización, protocolo NTP, protocolo PTP, SCADAs, EPICS, WinCC OA.

I. INTRODUCCIÓN

La sincronización de tiempo dentro de las infraestructuras críticas es una pieza fundamental para el correcto funcionamiento de la instalación. Esta red de sincronización es uno de los múltiples sistemas a monitorizar con el SCADA central escogido, que debe cumplir con varios requisitos de escalabilidad, seguridad, facilidad de uso e integración con los diferentes subsistemas.

Existen ejemplos en los que se puede ver la importancia de esta red de sincronización, como es el caso del proyecto IFMIF-DONES [1], donde se van a usar varios protocolos de sincronización de tiempo entre el CICS o sistema central y los múltiples LICS o sistemas locales [2].

Otro ejemplo es el radiotelescopio SKA [3], donde se pretende utilizar tecnología White Rabbit, ya que entre los

múltiples requerimientos, se necesita tener desfases de menos de 2 ns entre los dispositivos.

Por tanto, la sincronización de tiempo es, sin duda, un problema clave de las infraestructuras críticas que puede servir como caso de estudio para el análisis y comparación de sistemas de control a utilizar.

I-A. Protocolos de sincronización de tiempo

Para la sincronización de tiempo, existen tres tecnologías principales:

- **Protocolo NTP**, que permite sincronizar los dispositivos que no tengan grandes necesidades de precisión de tiempo para su funcionamiento [4]. Bajo NTP se controlan, por ejemplo, la mayoría de dispositivos domésticos como PCs o *smartphones*, y se consigue tener unos retardos entre el servidor y el cliente del orden de milisegundos.
- **Protocolo PTP**, que se encarga de la sincronización de dispositivos críticos y que permite una precisión del orden de los microsegundos [4].
- **Protocolo PTP con perfil de alta precisión (PTP-HA)**, basado en tecnología *White Rabbit* [5] y que necesita de hardware específico, por lo que no se ha tenido en cuenta para este estudio. Con esta tecnología se alcanzarían precisiones del orden de nanosegundos y picosegundos.

I-B. SCADAs para instalaciones científicas

Dentro del mundo de las instalaciones experimentales, como son los aceleradores de partículas o telescopios, se suelen utilizar sistemas de control distribuidos que son *open-source* y permiten un alto nivel de configuración, a costa de ser herramientas complejas de manejar. Además, estas herramientas, si bien funcionan muy bien como sistemas de control y comunicaciones entre dispositivos, sus capacidades como SCADA distan de ser las mejores, habiendo mejores alternativas en el mundo industrial.

Los dos sistemas de control distribuidos más conocidos dentro de este campo son EPICS [6] y Tango [7], si bien el sistema analizado en este estudio va a ser el primero de ellos. Se ha escogido EPICS porque dentro del proyecto ITER, donde se enmarca IFMIF-DONES, ya se está utilizando EPICS. Un ejemplo es el LIPAc, un prototipo de acelerador de partículas construido en Japón.

Desde el grupo de investigación LIPAc-DONES de la Universidad de Granada, se están proponiendo varias alternativas de SCADA industriales, con un funcionamiento más sencillo e integrado. Uno de los SCADAs que, a priori, parece más interesante es WinCC Open Architecture (OA) de Siemens [8], pues cumple con requisitos de seguridad SIL 3, permite el control de sistemas distribuidos con opciones de alta disponibilidad y balanceo de carga y es un sistema que ya ha demostrado su correcto funcionamiento dentro del CERN [9].

Por tanto, surge la necesidad de comparar ambas propuestas y para ello se ha realizado este primer estudio, donde se verán las principales características y se dará una primera valoración a cada una.

Concretamente, en la Sección II se describirá el setup experimental creado como caso de estudio para la comparación de ambos sistemas de control distribuido. En la Sección III y IV se analizarán, respectivamente, EPICS y WinCC OA, poniendo en valor cada una de sus principales características. Y por último, una vez examinados ambos sistemas, en la Sección V se discutirán los resultados obtenidos y en la Sección VI se darán unas conclusiones finales.

II. SETUP EXPERIMENTAL

II-A. Hardware utilizado

Para la comparación de los SCADAs se ha ideado un caso de estudio, donde se ha implementado una red de sincronización con múltiples nodos acorde a las necesidades que se pueden encontrar dentro de una infraestructura crítica.

En la propuesta realizada para este estudio se necesita un SCADA principal que deberá monitorizar los distintos subsistemas que forman parte de la infraestructura. Cada uno de estos subsistemas contará con distintos equipos que deberán de estar sincronizados en tiempo con el maestro PTP y el servidor NTP central de la instalación. Por tanto, a la hora de realizar la simulación, se debe de contar con un sistema que haga de servidor de tiempo y otros tantos que realicen el papel de nodos cliente.

Como se puede apreciar en Fig. 1, se han utilizado dos PCs donde van a funcionar, de manera simultánea, los dos SCADAs motivo de comparación. Por otro lado, los servidores de tiempo se encontrarán alojados en una Raspberry Pi 3 que va a ejecutar tanto el maestro PTP como el servidor NTP, ambos como demonios directamente en el sistema operativo anfitrión.

Para simular los distintos subsistemas a monitorizar, se ha hecho uso de una Raspberry Pi 4, una Jetson Nano de 2 GB y otra Jetson Nano de 4 GB. Dentro de cada uno de estos dispositivos, habrá un esclavo PTP y un cliente NTP ejecutándose en el sistema anfitrión. Además, en cada uno

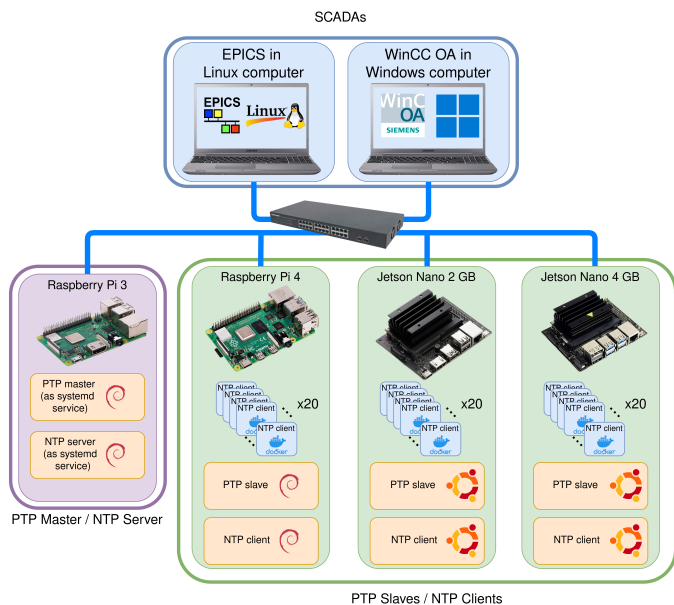


Figura 1. Esquema del setup experimental.

de estos equipos se han añadido 20 contenedores de Docker donde se ejecutarán nodos NTP adicionales.

De este modo, se encuentran un total de 3 subsistemas a monitorizar y un total de 63 nodos NTP y 3 clientes PTP que deberán de sincronizarse con las referencias de tiempo centrales.

II-B. Herramientas software

En cuanto a las herramientas software, lo primero que se encuentra son los sistemas de control distribuido a comparar en este artículo, EPICS y WinCC OA. Cada una de estas herramientas tienen filosofías de funcionamiento totalmente opuestas.

WinCC OA integra todos los elementos software necesarios para su funcionamiento como SCADA. Concretamente, el servicio de alarmas, el HMI (Human-Machine Interface) y el gestor de base de datos vienen incluidos con la instalación del software.

Sin embargo, el ecosistema de EPICS no funciona de la misma manera. EPICS Base solo incluye la capacidad de crear variables en cada uno de los dispositivos, mediante los llamados IOCs dentro de EPICS, así como también la comunicación entre ellos a través de un protocolo propietario *pvAccess*. De tal forma que el resto de las funcionalidades de un SCADA se han de añadir mediante otras herramientas, tal como se describirá en la Sección III.

Por otro lado, también se ha hecho uso de Docker para poder replicar más nodos NTP y tener un caso de uso más realista. Se intentó con PTP, si bien finalmente se descartó por haber problemas de incompatibilidad entre los contenedores. PTP es un protocolo más preciso que NTP y requiere de un sellado de los paquetes a través de la NIC del PC, lo que hace inviable correr varios esclavos PTP dentro de un mismo dispositivo.

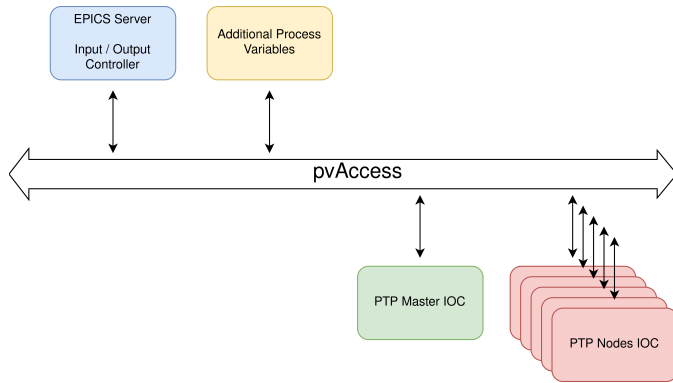


Figura 2. Diagrama de EPICS, el protocolo *pvAccess* permite comunicar todos los IOCs entre ellos.

III. EPICS

EPICS es el acrónimo de *Experimental Physics and Industrial Control System* y es un sistema de control distribuido que nació como un desarrollo del *Los Alamos National Laboratory* (LANL) en el año 1988.

El protocolo *pvAccess* [10] funciona bajo UDP y TCP. Concretamente utiliza UDP para descubrir los distintos IOCs, tras lo cual se conecta con ellos por TCP. Como se puede ver en Fig. 2, todos los dispositivos o IOCs comparten la misma información entre ellos. De esta manera, las variables, o *Process Variables* (PVs) como se denominan en EPICS, pueden ser leídas y modificadas por todos los dispositivos que se encuentran en la misma red.

Esta es la funcionalidad que ofrece EPICS Base, pero para poder dotar a EPICS de las capacidades típicas de los SCADAs, como por ejemplo de HMI, de almacenamiento de los datos o de gestión de alarmas, se debe recurrir a servicios de terceros desarrollados por la comunidad de EPICS.

III-A. Interfaz de usuario: CSS Phoebus

Dentro del ecosistema EPICS, CSS Phoebus es la herramienta más conocida para poder hacer interfaces de usuario o HMI, ver Fig. 3. Es una herramienta escrita en Java y que, además, permite conectarse con el resto de servicios de EPICS, como por ejemplo el *Archive Engine* para almacenar datos o el *Alarm Server* y el *Alarm Logger* para la gestión de alarmas. Además, permite la animación de los distintos elementos con scripts utilizando Jython, un intérprete de Python para Java.

Sin embargo, uno de los principales problemas de CSS Phoebus es que está pensado para entornos colaborativos donde todo el mundo puede editar y modificar la interfaz. Esto puede funcionar bien en instalaciones científicas más pequeñas, pero en el caso de infraestructuras más grandes como IFMIF-DONES, se requiere de un control de usuarios donde cada uno tenga un rol y unos determinados permisos, y donde no todos los usuarios puedan modificar la interfaz.

III-B. Almacenamiento de los datos: Archive Engine

Para el caso de los aceleradores de partículas, almacenar los datos que se produzcan tras cada uno de los experimentos

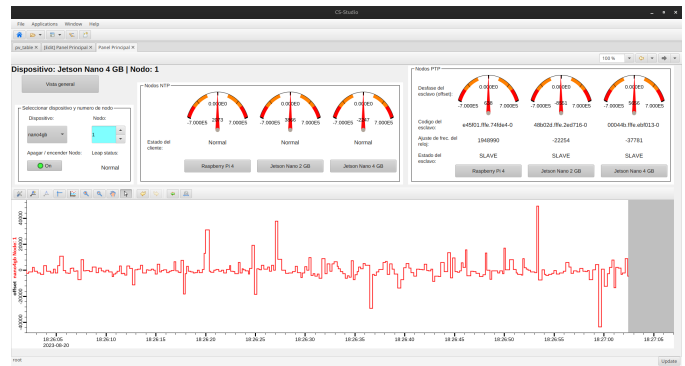


Figura 3. Ejemplo de interfaz en CSS Phoebus.

lanzados es uno de los aspectos clave de la instalación, puesto que posteriormente estos resultados serán utilizados por la comunidad científica para poder extraer conclusiones de ellos.

EPICS dispone de dos métodos distintos para almacenar los datos. El primero de ellos es el *Archive Appliance* [11], que no ha sido analizado en este estudio por su complejidad y su poca similitud con su homólogo en WinCC OA. Este servicio promete poder almacenar millones de PVs con una frecuencia muy alta de datos en cada uno de ellos. Su funcionamiento se basa en la creación de ficheros donde se almacenan los datos con tecnología ETL y codificados con Protobuf.

Por otro lado, se encuentra el *Archive Engine*, otro servicio que si permite almacenar datos en un SGBD tradicional. Este servicio tiene opciones para almacenar datos en PostgreSQL, Oracle y MySQL/MariaDB, que ha sido el SGBD utilizado en esta comparativa. Para poder almacenar los datos con este método, es necesario instalar y configurar a parte tanto MariaDB como el *Archive Engine*, que además necesitará una configuración adicional en CSS Phoebus para poder integrarse en la interfaz de usuario.

III-C. Servicio de alarmas: Alarm Server y Alarm Logger

Para poder disponer de alarmas en EPICS, es necesario instalar dos servicios distintos. El primero de ellos es el *Alarm Server* [12], que nos va a permitir crear un árbol de alarmas por dispositivos o áreas, reconocer las alarmas y crear distintas políticas como retardos en el disparo o actuar de manera automática realizando algún tipo de acción preventiva.

El otro servicio es el *Alarm Logger*, que nos va a permitir tener un registro de las alarmas, almacenando esta información en Elasticsearch [13] y permitiéndonos acceder a esa información con Kibana [14] o, directamente, con CSS Phoebus.

Por tanto, con estos dos servicios se puede tener un gestor de alarmas muy completo, aunque tiene el inconveniente de estar compuesto por demasiados programas que deben de funcionar en conjunto. En concreto, se necesita de una instalación de Apache Zookeeper [15], Apache Kafka [16], Elasticsearch y Kibana. De hecho, fue bastante difícil poner a funcionar todas estas herramientas en conjunto, teniendo que recurrir finalmente a una solución basada en Docker [17].

Finalmente, señalar también que en cuanto a los niveles de severidad para las alarmas, en EPICS solo hay tres tipos:

- “MINOR”: nivel de severidad bajo.
- “MAJOR”: nivel de severidad alto.
- “INVALID/UDF”: la variable a la que hace referencia no se encuentra disponible o tiene un valor inválido.

IV. WINCC OA

WinCC OA es una herramienta creada por la empresa ETM, que posteriormente fue absorbida por Siemens AG. Dentro de la gama de productos de Siemens, WinCC OA es el SCADA con las mayores prestaciones, utilizándose en infraestructuras enormes como en el CERN. Dispone de capacidades de distribución de carga y alta disponibilidad, y también soporta OPC UA [18], el estándar de facto en cuanto a protocolos de comunicación industriales. Por esta razón, en este estudio se ha utilizado OPC UA para comunicarse con la red de sincronización, usando el cliente OPC UA en el lado de WinCC OA y tres servidores OPC UA en cada uno de los tres dispositivos que albergan los nodos NTP y PTP (dispositivos en verde dentro del esquema de Fig. 1).

WinCC OA se vende como un sistema de control completo con capacidades SCADA. A lo largo de esta sección se podrá observar, por ejemplo, que integra todo lo necesario para poder tener una interfaz de usuario o almacenar los datos, incluyendo hasta el propio SGBD.

Dentro de WinCC OA hay que destacar su filosofía de funcionamiento, basado todo en *Datapoints*, que es la manera de declarar variables dentro del software y que permite añadirle atributos para poder almacenar su información o bien aplicarle alguna política para las alarmas.

IV-A. Interfaz de usuario

Para la creación de interfaces o HMI (ver Fig. 4), WinCC OA dispone de una herramienta muy completa y que permite un alto nivel de personalización. De hecho, este es una de sus principales ventajas y, a la misma vez, uno de sus pocos inconvenientes. Por ejemplo, para el uso de algunos widgets, como un simple *slider*, hay que programar todo el funcionamiento del mismo con su lenguaje de programación propio llamado Ctrl [19], que tiene ciertas similitudes con C y que soporta programación orientada a objetos.

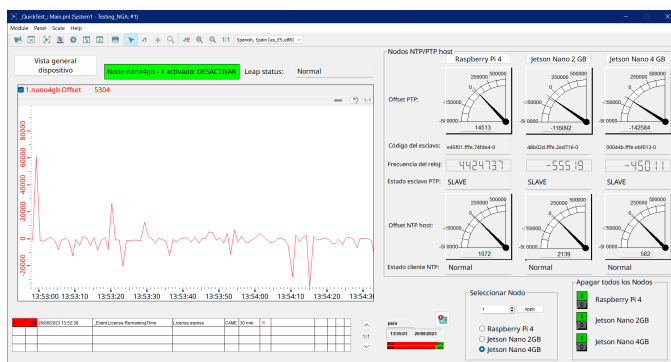


Figura 4. Ejemplo de interfaz en WinCC OA.

Otra de las ventajas es que, a diferencia de CSS Phoebus, en WinCC OA el entorno de desarrollo para crear y modificar los paneles HMI y el programa que visualiza los mismos está separado. Para un despliegue dentro de una infraestructura crítica es un aspecto muy importante, pues los usuarios no pueden tener la libertad para modificar los paneles HMI, tarea reservada para los técnicos e ingenieros.

IV-B. Almacenamiento de los datos

Como ya se ha comentado al inicio de esta sección sobre WinCC OA, en la propia instalación se incluye el SGBD que se necesite, no teniendo que preocuparnos por su despliegue.

WinCC OA dispone de varios SGBDs. El elegido ha sido InfluxDB, pues es el que viene por defecto y el que sería más adecuado para el caso de uso de la red de sincronización, ya que InfluxDB está especializado en almacenar datos de series temporales. Otros SGBDs disponibles son PostgreSQL, MySQL/MariaDB y Oracle, aunque se ha de pagar una licencia extra por cada uno de ellos.

Otra de las ventajas en este apartado es el hecho de la fácil configuración del SGBD, ofreciéndonos muchas opciones para realizar copias de seguridad automáticas basadas en distintas políticas, guardar datos en la nube o establecer periodos de retención; todo esto dentro del mismo SCADA y sin tener que recurrir a herramientas externas.

IV-C. Servicio de alarmas

Las alarmas en WinCC OA tienen más niveles de severidad que en EPICS, teniendo hasta 7 clases por defecto más las que puede añadir el usuario. Además, existen varios paneles predefinidos para el control de las alarmas y se pueden añadir comentarios o ejecutar acciones cuando se dispare alguna de ellas.

No obstante, en WinCC OA no hay más remedio que usar InfluxDB para almacenar el histórico de alarmas, no permitiendo cambiar a otro SGBD como si ocurre con la adquisición de datos.

V. RESULTADOS Y DISCUSIÓN

Una vez realizada sendas aplicaciones de monitorización utilizando cada uno de los sistemas de control propuestos, y teniendo en cuenta las principales características de cada uno de ellos descritas en el apartado anterior, a continuación se procede a compararlos:

V-A. Instalación e integración

Como se ha podido ver, EPICS y WinCC OA son sistemas con filosofías totalmente opuestas. El primero es un software libre, cuya principal baza es “ofrecer poco para limitar poco”, mientras que en el segundo ya viene todo incluido para que el ingeniero solo se tenga que preocupar por empezar a crear la aplicación de monitorización.

Todo en EPICS es configurable y adaptable a nuestras necesidades, pero justo eso es lo que lo hace difícil y tedioso para la gran mayoría de servicios.

El protocolo *pvAccess* es muy cómodo ya que, tan pronto como tengas EPICS instalado en tu dispositivo, se puede

empezar a publicar sus PVs en la red. Pero para ello debes de integrar EPICS en tus dispositivos de alguna manera al no ser un protocolo estandarizado en la industria, lo que obliga a acoplar algún Linux empotrado al lado de un autómatas programable (PLC) o un robot industrial.

Por estas razones, la solución de Siemens parece más adecuada, ya que su instalación es bastante más fácil y además soporta protocolos de comunicación estándares como OPC UA, Ethernet/IP, MQTT, etc., de forma nativa.

V-B. Interfaz HMI

En cuanto a capacidades HMI, los dos sistemas permiten realizar aplicaciones bastante similares, como se puede ver en Fig. 3 y Fig. 4. Con ambos SCADAs se ha conseguido realizar las aplicaciones de monitorización sin mayor problema. Las dos interfaces realizadas tienen una funcionalidad similar y no ha habido ninguna acción que en un sistema no se haya podido implementar y en el otro sí.

Por un lado, CSS Phoebus es muy fácil e intuitivo de utilizar, además de que los gráficos que aporta son más claros y modernos. La integración de Python a través de Jython permite interactuar con los objetos en un lenguaje de programación de sobras conocido, lo que es un gran valor añadido.

A todo esto hay que sumarle que las tareas de visualización más triviales, como atribuir el valor de una variable a un *slider*, no requiere de programación previa. Al contrario que en WinCC OA, donde para todo hay que recurrir a su lenguaje Ctrl.

Sin embargo, da la sensación de que CSS Phoebus, a la larga, pueda quedarse limitado en cuanto a opciones, ya que dispone de menos objetos a añadir. No obstante, y teniendo en cuenta el estudio realizado hasta ahora, el HMI de EPICS se lleva la victoria en este apartado.

V-C. Alarmas

EPICS y WinCC OA también son parecidos en cuanto a la gestión de las alarmas. Ambos sistemas ofrecen bastantes opciones en cuanto a registrar alarmas, reconocerlas, etc. No obstante, hay varios factores que marcan la diferencia.

El primero de ellos es la severidad y las clases de alarmas disponibles, WinCC OA tiene muchas más que EPICS, que solo dispone de dos categorías (*MINOR* o *MAJOR*). Lo que es un punto a favor para WinCC OA.

Otro factor es el SGBDs utilizado para registrar las alarmas, en EPICS se usa Elasticsearch, mientras que en WinCC OA solo se puede usar InfluxDB.

InfluxDB es una fantástica base de datos para almacenar series temporales, pero no es la indicada para guardar *logs* de errores; tarea en la que si está especializado Elasticsearch. Esto le da cierta ventaja a EPICS.

Sin embargo, en términos generales, WinCC OA parece tener mejores soluciones en cuanto a la gestión de alarmas, ya que ofrece más clases, más opciones, la capacidad de aplicar niveles de prioridad y su integración con los *Datapoints* es un gran añadido, a pesar de no usar el mejor SGBDs para guardar los *logs*.

V-D. Almacenamiento de los datos

Ambos SCADAs tienen compatibilidad con MySQL/MariaDB, PostgreSQL y Oracle a la hora de registrar los datos de las variables. Además, WinCC OA también cuenta con InfluxDB, que para los datos de la red de sincronización es el SGBD más adecuado.

Las opciones disponibles son bastante similares en términos generales, si bien WinCC OA va un paso más allá al ofrecer la capacidad de crear copias de seguridad en la nube y tener periodos de retención. Con EPICS se podrían conseguir las mismas capacidades, pero habría que depender de herramientas ajenas al SCADA.

Por tanto, si bien ambos sistemas constan de capacidades similares, WinCC OA ofrece unas cuantas opciones más que hacen que, de nuevo, la balanza se decante a favor de este.

V-E. Control de usuarios y despliegue de la aplicación

En el apartado III-A se pudo constatar como EPICS adolece de no tener un control de usuarios en su interfaz. A esto se suma que tanto el editor de la interfaz como el despliegue de la misma se hace con el mismo paquete sin diferenciar entre ambos, lo que permite al usuario modificar la interfaz sin tener permisos para ello.

El problema es aún peor cuando se compara con WinCC OA, que dispone de la capacidad de crear varios usuarios, otorgarle hasta 16 niveles de permisos distintos y un gestor de áreas. Para dotar a EPICS del gestor de áreas, habría que recurrir a un módulo adicional llamado *areaDetector* [20].

Por tanto, este punto hace que WinCC OA sea la única alternativa viable para infraestructuras críticas de gran calibre como IFMIF-DONES o el radiotelescopio SKA.

V-F. Tabla resumen

A continuación, se muestra una tabla resumen con las valoraciones de cada SCADA en cada uno de los apartados:

Cuadro I
TABLA RESUMEN CON LAS VALORACIONES DE CADA SCADA

Apartado	EPICS	WinCC OA
Instalación e integración	5	10
Interfaz HMI	9	8
Alarmas	6	9
Almacenamiento de los datos	7	9
Control de usuarios y despliegue	1	9
PUNTUACIÓN FINAL	5,6	9

VI. CONCLUSIONES

En este trabajo se ha propuesto una red de sincronización entre dispositivos, simulando varios nodos NTP y obteniendo así un caso de estudio típico en infraestructuras críticas.

Con esta red de sincronización como base, se han desarrollado dos aplicaciones SCADA con dos de los sistemas de control distribuido que actualmente se están utilizando o proponiendo en las arquitecturas de control y supervisión de varios aceleradores de partículas como son el CERN o el futuro IFMIF-DONES y también en radiotelescopios como

el SKA. Basándonos en la experiencia adquirida desarrollando las aplicaciones se han podido analizar algunas de las principales funcionalidades de los SCADAs y proporcionar unas evaluaciones preliminares, concluyendo que según esta primera aproximación parece más adecuado desde el punto de vista técnico utilizar un SCADA industrial, como es WinCC OA, frente a la solución tradicional utilizando sólo EPICS.

Como se ha podido ver a lo largo de este estudio, el principal problema de EPICS es que es un buen sistema de control distribuido pero sus funcionalidades SCADA no son tan buenas. EPICS es un sistema cuya fiabilidad y escalabilidad está más que probada por muchos proyectos, entre ellos el proyecto ITER [21] o el LIPAc [22]. Por tanto, el uso de EPICS a bajo nivel con su protocolo *pvAccess* puede ser compatible con el uso de un SCADA industrial en un nivel superior. La única cuestión a resolver sería la comunicación entre ambos sistemas de control. Como OPC UA es el protocolo de comunicación industrial más estandarizado, se podría proponer su uso en el lado del SCADA industrial. Ya existen proyectos de *drivers* OPC UA para EPICS [23].

Estudiar el caso de la sincronización de tiempo ha permitido analizar las funciones básicas de un SCADA. No obstante, como trabajo futuro derivado de este estudio se propone la implementación real de un prototipo del acelerador IFMIF-DONES, donde se puedan evaluar otras importantes funcionalidades como son la seguridad, la fiabilidad o la escalabilidad del sistema de control y monitorización para una gran infraestructura como esta.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto PID2021-123930OB-C22 del MCIN/ AEI /10.13039/501100011033/ y por FEDER “Una manera de hacer Europa” y parcialmente por la Ayuda para la Formación de Profesorado Universitario (FPU): FPU20/1857.

Así mismo, agradecer también el apoyo para este trabajo del Departamento de Ingeniería de Computadores, Automática y Robótica (ICAR) de la Universidad de Granada, así como también del Centro de Investigación en Tecnologías de la Información y la Comunicación (CITIC-UGR).

REFERENCIAS

- [1] C. Megías, V. Vázquez, E. Ros, M. Cappelli, J. Díaz. “Ethernet-based timing system for accelerator facilities: The IFMIF-DONES case,” *Computer Networks*, vol. 233, 2023, 109897, ISSN 1389-1286, <https://doi.org/10.1016/j.comnet.2023.109897>
- [2] M. Cappelli, F. Ambí, A. Bagnasco, E. Botta, Z. Chen, J. Diaz, V. Gutierrez, P. Goryl, J. Sousa, A. Ibarra. “Recent advances of the IFMIF-DONES central instrumentation and control systems engineering design,” *Fusion Engineering and Design*, vol. 194, 2023, 113671, ISSN 0920-3796, <https://doi.org/10.1016/j.fusengdes.2023.113671>
- [3] M. Jiménez-López, F. Torres-González, J. L. Gutiérrez-Rivas, M. Rodríguez-Álvarez, J. Díaz, “A Fully Programmable White-Rabbit Node for the SKA Telescope PPS Distribution System,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 2, pp. 632-641, Feb. 2019, <https://doi.org/10.1109/TIM.2018.2851658>
- [4] T. Neagoie, V. Cristea, L. Banica, “NTP versus PTP in Computer Networks Clock Synchronization,” 2006 IEEE International Symposium on Industrial Electronics, Montreal, QC, Canada, 2006, pp. 317-362, <https://doi.org/10.1109/ISIE.2006.295613>
- [5] M. Lipiński, T. Włostowski, J. Serrano, P. Alvarez, “White rabbit: a PTP application for robust sub-nanosecond synchronization” 2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, Munich, Germany, 2011, pp. 25-30, <https://doi.org/10.1109/ISPCS.2011.6070148>
- [6] Experimental Physics and Industrial Control System. Visitado por última vez: 12 de Marzo de 2024. Disponible: <https://epics.anl.gov/>
- [7] Tango Consortium. Visitado por última vez: 12 de Marzo de 2024. Disponible: <https://www.tango-controls.org/>
- [8] Siemens AG. SIMATIC WinCC Open Architecture. Visitado por última vez: 13 de Marzo de 2024. Disponible: <https://www.siemens.com/global/en/products/automation/industry-software/automation-software/scada/simatic-wincc-oa.html>
- [9] P. Golonka, W. Fabian, M. Gonzalez-Berges, P. Jasiun, F. Varela-Rodriguez, “FwWebViewPlus: integration of web technologies into WinCC OA based Human-Machine Interfaces at CERN,” *Journal of Physics: Conference Series*, vol. 513, No. 1, p. 012009, <https://doi.org/10.1088/1742-6596/513/1/012009>
- [10] M. Sekoranja, M. Kraimer, G. White, A. Johnson, B. Franksen, M. Abbott, P. Duval, “pvAccess Protocol Specification,” <https://epics-control.s.org/wp-content/uploads/2018/10/pvAccess-Protocol-Specification.pdf>
- [11] M. Shankar, L. F. Li, M. A. Davidsaver, M. G. Konrad, (2015, October). “The EPICS archiver appliance,” *Proceedings of ICALEPCS*, vol. 15, <https://accelconf.web.cern.ch/ICALEPCS2015/papers/wepgf030.pdf>
- [12] K. Kasemir, K. Schroff, “Phoebus Alarm System: Alarm Server”, June 2019 EPICS Meeting, https://indico.cern.ch/event/766611/contribution/s/3438293/attachments/1854259/3045083/Phoebus_Alarm_System.pdf
- [13] Elasticsearch: Motor de búsqueda y analítica distribuido oficial. Visitado por última vez: 12 de Marzo de 2024. Disponible: <https://www.elastic.co/es/elasticsearch>
- [14] Kibana: Explora, visualiza y descubre datos — Elastic. Visitado por última vez: 12 de Marzo de 2024. Disponible: <https://www.elastic.co/es/kibana>
- [15] Welcome to Apache ZooKeeper™. Visitado por última vez: 12 de Marzo de 2024. Disponible: <https://zookeeper.apache.org/>
- [16] Apache Kafka. Visitado por última vez: 12 de Marzo de 2024. Disponible: <https://kafka.apache.org/>
- [17] docker-phoebus-alarm. Visitado por última vez: 12 de Marzo de 2024. Disponible: <https://github.com/giosava94/docker-phoebus-alarm>
- [18] M. H. Schwarz, J. Börsök, “A survey on OPC and OPC-UA: About the standard, developments and investigations,” 2013 XXIV International Conference on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 2013, pp. 1-6, <https://doi.org/10.1109/ICAT.2013.6684065>
- [19] Introduction to CTRL. WinCC OA 3.19 Documentation. Visitado por última vez: 12 de Marzo de 2024. Disponible: https://www.wincco.com/documentation/WinCCOA/latest/en_US/Control_Grundlagen/Control_Grundlagen.html
- [20] M. Rivers, “areaDetector: Software for 2-D Detectors in EPICS,” *AIP Conference Proceedings*, vol. 1234, No. 1, pp. 51-54, American Institute of Physics, <https://doi.org/10.18429/JACoW-ICALEPCS2017-THDPL03>
- [21] ITER Fusion Device. EPICS Controls. Visitado por última vez: 17 de Mayo de 2024. Disponible: <https://epics-controls.org/projects-archive/iter/>
- [22] J. Calvo, M. L. Rivers, M. A. Patricio, A. Ibarra (2012). “EPICS based low-level radio frequency control system in LIPAc,” *Fusion Engineering and Design*, 87(11), 1872-1879. <https://doi.org/10.1016/j.fusengdes.2012.09.013>
- [23] R. Lange, R. Elliot, B. Kuner, K. Vestin, C. Winkler, D. Zimoch (2021, October). “Integrating OPC UA Devices in EPICS,” In 18th International Conference on Accelerator and Large Experimental Physics Control Systems (p. MOPV026). <https://doi.org/10.18429/JACoW-ICALEPCS2021-MOPV026>