

让PI系统发掘企业数据蕴藏的无限潜能



# High Availability 提高 PI 系统的可用性

#### 李捍永,技术工程师



VALUE NOW, VALUE OVER TIME

### Introduction

High Availability (HA)

#### "Ability of a system to tolerate faults and continue to provide service according to its specifications"

Dr. Kalinsky "Design Patterns for High Availability"

#### Objective of the Day

- Review the HA architecture
- Benefits and Limitations
- Implementing a HA PI server
- Upcoming Enhancement

**OSI**soft



### HA architecture



#### **OSI**soft.

#### VALUE NOW, VALUE OVER TIME

## Values of HA

- High Availability to your PI System
- Peace of mind for Administrators
- Direct support for existing PI Clients
- Simple, scalable and flexible architecture



# Limitations of HA

- No automatic replication of non-interface data
- No replication of batch records
- Post processed data calculated independently
- PI ACE management requires primary

## Requirements

- The servers
  - Standard hardware and operating system is all that's required
- Upgrade underlying PI-API & PI-SDK on your interface nodes
- Consider upgrading your interface to the latest version
- Use SMT or ICU on the interface node to setup and monitor the test

# Upgrade and Install PI

- Start your upgrades at the PI server
  - Prepare the server (O/S, network, etc)
    - Use documented procedure for moving PI to a new server
  - Upgrade PI to latest version
  - Install latest PI on secondary machine
  - Use PI collective manager to promote your PI clone into a collective with one member



# Using Collective Wizard

• Demo Video



VALUE NOW, VALUE OVER TIME

# **Configure Interface**

- If the interfaces aren't already on interface nodes, move them off of the PI server
- Upgrade PI-API & PI-SDK

**OSI**soft

- Optionally upgrade your interface(s) to latest version
- Set up N-Way buffering using latest PI-API Buffering version

### **Enable Buffer**

Choose Buffer Type Buffering Settings Buffered Servers PI Buffer Subsystem Service <sup>i</sup> Parameter Details API Buffer Server Service	<ul> <li>Buffering allows continuous collection of data on an API Node regardless of the status of the PI server or the network link to the server.</li> <li>Disable buffering</li> <li>Enable buffering with PI Buffer Subsystem Service status: Stopped Startup type: Automatic Number of dependent services: 1</li> <li>Number of running dependent services: 0</li> </ul>
	<ul> <li>Enable buffering with API Buffer Server</li> <li>Service status: Stopped</li> <li>Startup type: Disabled</li> <li>Number of dependent services: 0</li> <li>Number of running dependent services: 0</li> </ul>

# **Buffer Settings**

#### **API Buffer**

Choose Buffer Type

**Buffering Settings** 

Buffered Servers

**OSI**soft.

API Buffer Server Service

Buffer and Replicate using the following configuration:

Click once in the Buffered or Replicated column to toggle between On and Off.

Add a server	:
--------------	---

Server	Buffered	Replicated
HANYONGD610	Yes	Yes
congd400	No	No
hycollective2nd	Yes	Yes

#### PI Buffer Subsystem

Choose Buffer Type	Buffering to collective/server: hanyongd610											
Buffering Settings	Replicate data to all collective member podes											
Suffered Servers	I♥ Replicate data to all collective member hodes											
PI Buffer Subsystem Service	Buffered Server Names											
Parameter Details	Path C Name C IP Address											
API Buffer Server Service												
	Server Collective Member Type											
	☑ hanyongd610.osisoft.int hanyongd610 Primary											
	I 192.168.85.47 hanyongd610 Secondary											

#### VALUE NOW, VALUE OVER TIME

Add Server

### A Timeline

Obtain & Prepare Test												
Server(s)												
Clone existing PI server												
Install PI on test server												
Backup Existing PI server												
Overlay files, run utilities on test												
server												
Upgrade test server												
Create Collective using test												
server												
Prepare interface nodes												
upgrade PI-API & PI-SDK												
Upgrade interface(s)												
Set up N-Way Buffering												
The Test												
Confirm buffering to both												
servers	 _											
copy and test existing												
applications	 _	 						 				
key users access test server	 _	 										
venty local procedures	 _	 <u> </u>							<u> </u>	<u> </u>		
resolve issues with USI												
assistance	 _	 	 		$\vdash$	 						
7 1 0 1 1 1 1	 _	 	 		$\vdash$		 					
lest Successful? Next steps	 _	 	 		$\vdash$							
Uninstall PI from test server(s)	 _	 			$ \rightarrow $	 		 				
Make a new clone from existing												
server	 _	 										
make your collective	 	 										
point users at new collective	 _	 										
remove old server from												
buffering on interface nodes												

OSIsoft.

## Upcoming enhancement



#### OSIsoft.

VALUE NOW, VALUE OVER TIME

### Conclusion

- Implement HA for:
  - improve reliability
  - better
     manageability

**OSI**soft.

