

```
import {ThemeProvider} from '@material-ui/core/styles';
import CssBaseline from '@material-ui/core/CssBaseline';
import {Container} from '@material-ui/core';
import {useApollo} from '../graphql/client';

import {lightTheme, darkTheme} from '../utils/theme';
import useLocalStorage from '../hooks/useLocalStorage';

import NavBar from '../components/NavBar';

function App({Component, pageProps}: AppProps) {
  const [currentTheme, setCurrentTheme] = useLocalStorage('key:theme-value', initialValue:'light');
  const apolloClient = useApollo(pageProps.initialApolloState);

  useEffect(() => {
    const jssStyles = document.querySelector('selector: #jss-server-side');
    if (jssStyles) {
      jssStyles.parentElement.removeChild(jssStyles);
    }
  }, [deps]);
}

return (
  <>
  <Head>
    <title>ECU-DEV</title>
    <meta name="viewport" content="minimum-scale=1, initial-scale=1, width=device-width, height=device-height, user-scalable=no">
  </Head>
  <Component {...pageProps} theme={currentTheme === 'light' ? lightTheme : darkTheme}>

```

PSUSU

1. Pogodnosti i primena distribuiranih upravljačkih sistema

Pogodnosti

Primena

Uvodjenje

2. Hijerarhijska arhitektura distribuiranih upravljačkih sistema (funkcionalna dekompozicija)

Nivoi

3. Fizička i funkcionalna arhitektura sistema

Funkcionalna (Pitanje iznad)

Fizicka

Komunikacione Mreze

Primena

4. Definicija i arhitektura klasičnog SCADA sistema

Komponente

Karakteristike

5. Arhitektura RTU i komunikacija sa RTU

Arhitektura

Komunikacija

6. Osnovni zadaci (podsistemi) SCADA sistema

1. Akvizicija (definisanje veličina)

2. Alarmi

3. Trendovi

4. Komandovanje (recepti)

5. Izvestaji

6. Prikaz Stanja (Grafika)

7. Komunikacioni Podsystem (Uredjaji)

8. Skladistenje

9. Mrežni Podistem

9. Distribuirani računarski sistem (definicija, primeri primene)

Primeri

Razlozi Pojave

10. Osobine Distribuiranog Sistema

Pozitivne Osobine
Negativne Osobine (U odnosu na centralizovane)
Middleware (Nije deo pitanja)

11. Ciljevi Distribuiranog Sistema
Povezivanje Resursa sa Korisnicima
Transparentnost
Otvorenost
Skalabilnost
Sigurnost

12. Tipovi distribuiranih sistema (Klaster, Grid Computing)
Klaster Sistemi
Grid Computing
Razlike

13. 5 Koraka Omron i IFIX SCADA Sistema
1. Konfiguracija i Postavljanje
2. Kreiranje Baze Tagova
3. Programiranje i Razvoj Logike
4. Dizajn HMI (Human-Machine Interface)
5. Testiranja i Validacija
Random 💀 Nekoherentne 💀 Beleske

14. OPC (+WinCC)
Specifikacije unutar OPC standarda
DA specifikacija
HDA specifikacija
WinCC

15. Cloud
Definicija Cloud Computinga
Prednosti Cloud Computinga
Ekonomija velikih brojeva
Pay as you go (Plaćanje po upotrebi)
Niska početna cena
Visoka dostupnost
Nivoi usluga u Cloud-u
Najveći Cloud Provajderi
Virtuelne Mašine
Prednosti Clouda
Data Storage
Nivoi Usluga u Cloud Computingu

1. Pogodnosti i primena distribuiranih upravljačkih sistema

Distribuirani upravljački sistem (DRUS) je savremena tehnologija koja omogućava decentralizovano upravljanje industrijskim procesima kroz mrežu međusobno povezanih računara i kontrolnih jedinica. Ovi sistemi koriste napredne metode automatizacije i komunikacione tehnologije kako bi omogućili efikasno i fleksibilno upravljanje kompleksnim procesima u različitim industrijama.

Ne treba brkati sa distribuiranim racunarskim sistemima! DRUS je samo podskup DRS-a

Pogodnosti

Povećanje produktivnosti proizvodnje: Automatizacija smanjuje ljudske greške i optimizuje radne procese.

Poboljšanje kvaliteta proizvoda: Precizna kontrola procesa osigurava konstantan kvalitet proizvoda.

Unapređenje sigurnosti i raspoloživosti postrojenja: Automatizovani sistemi brže reaguju na anomalije, smanjujući rizik od havarija.

Veća fleksibilnost rada postrojenja: Brže prilagodbe promenama u proizvodnom procesu.

Kvalitetniji uvid u rad postrojenja: Omogućava detaljnu analizu i optimizaciju proizvodnje.

Primena

Procesnim industrijama: Hemijska, petrohemidska postrojenja, rafinerije, cementare, železare, prehrambena industrija, vodosnabdevanje i tretman otpadnih voda.

Proizvodnim postrojenjima: Automobilска, elektronska i ostale proizvodne industrije.

Energetskim postrojenjima: Elektrane i obnovljivi izvori energije.

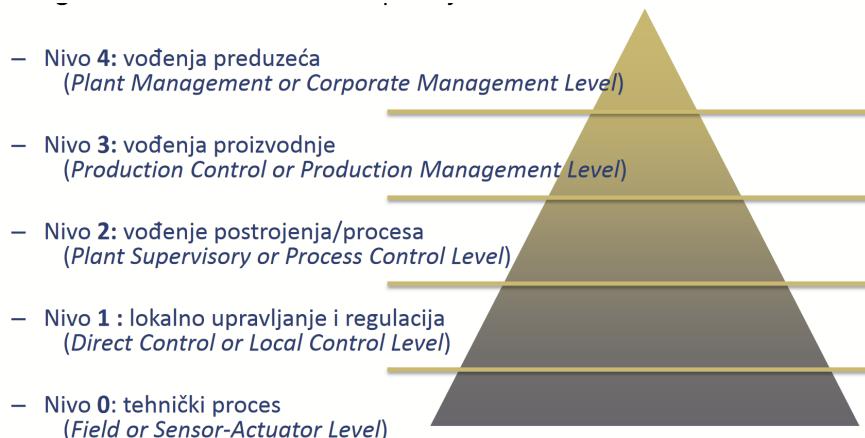
Laboratorijskim postrojenjima: Automatizacija složenih laboratorijskih procesa.

Uvodjenje

Nova proizvodna postrojenja: Dizajnirana s najnovijim tehnologijama upravljanja.

Revitalizacija postojećih postrojenja: Unapređenje starih sistema radi povećanja efikasnosti i produžetka životnog veka.

2. Hjerarhijska arhitektura distribuiranih upravljačkih sistema (funkcionalna dekompozicija)



Omogućava funkcionalnu dekompoziciju kompleksnih industrijskih sistema na nekoliko međusobno povezanih nivoa. Ova dekompozicija pomaže u efikasnijem upravljanju različitim aspektima proizvodnje, od tehničkih do menadžerskih zadataka.

Nivoi

- **Nivo 0: Tehnički proces**

Najniži nivo, **fokusiran na signale i uređaje**. U ovom sloju se **ne vrši upravljanje**, već se bavi **prikupljanjem podataka** o stanju i funkciji opreme. PLC (Programmable Logic Controller) je neophodan za upravljanje na ovom nivou kao veza sa visim nivoima.

- **Nivo 1: Lokalno upravljanje i regulacija**

Ovaj nivo koristi jednostavne algoritme poput ON-OFF, i PID za automatizovano **lokalno upravljanje**. Akvizicija procesnih veličina, nadzor i provera ispravnosti sistema vrše se autonomno, bez direktnog ljudskog učešća.

- Akvizicija procesnih veličina - prikupljanje trenutnih vrednosti mernih veličina i stanja komponenata postrojenja (npr. stanja pumpi, ventila, motora i sl.). Neophodna za upravljanje procesom u otvorenoj i/ili zatvorenoj petlji, nadzor procesa, izradu izveštaja o stanju procesa
- Nadzor procesa/postrojenja i provera ispravnosti sistema - procesiranje prikupljenih podataka, proveravanje njihove prihvatljivosti, donošenje odluka o akcijama koje treba preduzeti, provera funkcionalnosti računara i periferija, alarmiranje, dojavljivanje grešaka i neispravnih stanja.
- Sekvencijalno upravljanje i upravljanje u zatvorenoj petlji

- **Nivo 2: Vođenje postrojenja/procesa**

Uključuje **centralizovano real-time upravljanje**. Ovaj nivo se bavi optimalnim uslovima procesa, nadzorom, skladištenjem i izveštavanjem. Informacije se koriste za prilagođavanje parametara PLC-a i mikrokontrolera za optimalno upravljanje.

Implementiraju se funkcije:

- **Optimalno upravljanje procesom** - Optimizacija se sprovodi na osnovu matematičkog modela procesa, a prema nekom kriterijumu optimalnosti koji treba da osigura optimalan rad procesa/postrojenja u promenljivim radnim uslovima.
- **Adaptivno upravljanje** - Na osnovu merenih vrednosti procesnih veličina estimiraju se parametri matematičkog modela procesa iz kojih se zatim izračunavaju optimalne vrednosti parametara regulatora.
- **Optimalna koordinacija rada postrojenja** - Sprovodi se na osnovu: produktivnosti proizvodnje, stanja sirovina, stanja skladišta proizvedene robe, cene energije i dodatnih kriterijuma optimalnosti.
- **Nadzor, skladištenje i izveštavanje** - Nadzor performansi postrojenja, skladištenje podataka i izveštavanje o stanju.

- **Nivo 3: Vođenje proizvodnje**

Fokusira se na planiranje redosleda proizvodnje (production scheduling - promena narudžbi kupaca, stanja zaliha, energetskih ograničenja i zahteva.) prema raspoloživim resursima i nadzire celokupno proizvodno postrojenje. Na ovom nivou, koji **nije real-time**, inženjeri koriste sofisticirane metode za dnevnu optimizaciju proizvodnih operacija.

- **Nivo 4: Vođenje preduzeća**

Najviši nivo hijerarhijske strukture, bavi se nedeljnim i mesečnim planiranjem, uključujući **ekonomске, komercijalne i kadrovske aspekte**. Planiranje na ovom nivou nije u realnom vremenu i **ne utiče direktno na proizvodne operacije**.

Minimalna SCADA konfiguracija mora obuhvatati nivoe 0, 1 i 2 za osnovno upravljanje, dok su nivoi 3 i 4 posvećeni menadžmentu i imaju indirektni uticaj na proizvodnju.

3. Fizička i funkcionalna arhitektura sistema

Funkcionalna (Pitanje iznad)

Funkcionalna arhitektura opisuje **logičku organizaciju i raspodelu zadataka unutar sistema**. Ova arhitektura razlaže sistem na hijerarhijske slojeve, gde svaki sloj ima određene funkcije i odgovornosti. Funkcionalni slojevi omogućavaju specijalizaciju u upravljanju na različitim nivoima procesa, od lokalnog upravljanja do upravljanja celokupnim preduzećem.

Fizicka

Odnosi se na **konkretnu hardversku implementaciju** sistema. Ova arhitektura je često sabirnički orijentisana, sve komponente hardvera komuniciraju kroz zajednički komunikacijski kanal ili sabirnicu. **Sabirница** služi kao centralna tačka za razmenu podataka između različitih delova sistema. **Fizička integracija funkcija dva ili više funkcionalnih nivoa** u jedan fizički sloj može dovesti do pojednostavljenja distribuiranog sistema, smanjenja troškova i olakšanja održavanja. Primer bi bio dizajniranje servera koji ne samo da obrađuje podatke, već ima i integrisane kapacitete za skladištenje i specijalizovane alate za upravljanje mrežom. Ovo omogućava smanjenje fizičkog prostora potrebnog za sistem, smanjenje kompleksnosti upravljanja i održavanja, kao i troškova.

Komunikacione Mreze

- **Fieldbus:**
 - Povezuje **upravljačke uređaje sa senzorima** i izvršnim organima.
 - Omogućava preciznu **lokalnu kontrolu** i brzu razmenu podataka.
- **Upravljačka mreža na nivou postrojenja:**
 - Povezuje **više upravljačkih uređaja i SCADA sistem**.
 - Omogućava koordinaciju i nadzor na višem nivou postrojenja.
- **Poslovna magistrala:**
 - Povezuje **upravljački sistem sa ostalim poslovnim aplikacijama**.
 - Omogućava integraciju podataka između proizvodnih i poslovnih operacija.

Primena

U manjim proizvodnim postrojenjima, fizička arhitektura može se svesti na dva osnovna nivoa:

1. **Niži slojevi:** Fokusirani na lokalno upravljanje (automatika na nivoima 0, 1 i 2).
2. **Viši slojevi:** Kombinacija vođenja postrojenja i proizvodnje.

4. Definicija i arhitektura klasičnog SCADA sistema

SCADA (Supervisory Control And Data Acquisition) sistemi omogućavaju efikasno upravljanje procesima kroz **centralizovano nadzorno mesto**, čime se postiže blagovremeno (**real time**) i pouzdano dobavljanje podataka.

Komponente

- **RTU (Remote Terminal Unit):**

- RTU uređaji se nalaze na terenu i direktno prikupljaju podatke iz industrijskih uređaja kao što su senzori i aktuatori.
- RTU-ovi mogu biti zasnovani na PLC tehnologiji ili na standardnim mikrokontrolerima.
- Povezani su sa centralnom ili master stanicom putem različitih komunikacionih linija, uključujući žičane i bežične tehnologije.

- **Master Stanica:**

- Centralni deo SCADA sistema koji trajno čuva podatke prikupljene od strane RTU-ova.
- Omogućava vizualizaciju i analizu prikupljenih podataka, te upravljanje operacijama na terenu.
- Operateri koriste master stanicu za daljinsko upravljanje procesima i reagovanje na alarmne situacije.

Najbitnija komponenta SCADA sistema je **kontrolna soba**, gde se centralizuje upravljanje. Svaka greška na ovom nivou može imati značajne posledice, zbog čega je svaki aspekt sistema - od hardvera do softvera - dizajniran sa visokom stopom redundancije.

Karakteristike

- **Real time:** Sposobnost da brzo i u realnom vremenu prikuplja i obradi podatke.
- **Tačnost:** Visoka preciznost u prikupljanju podataka i pouzdanost prikaza.
- **Visoka pouzdanost:** U SCADA sistemima, pouzdanost se postiže dupliranjem ključnih komponenti, uključujući hardver, softver, komunikacione mreže i operativne sisteme.
- **Distribuirana arhitektura:** Omogućava široko geografsko pokrivanje, često koristeći wireless tehnologiju za komunikaciju između komponenti.

5. Arhitektura RTU i komunikacija sa RTU

RTU (Remote Terminal Unit) je ključni deo sistema nadzora i upravljanja, optimizovan za rad u **zahtevnim industrijskim uslovima**. Struktura RTU-a slična je standardnom računarskom sistemu, ali uključuje specijalizovane komponente za pouzdan rad u ekstremnim uslovima.

Arhitektura

1. **Osnovni Racunarski Elementi:** Napajanje, Procesor, Memorija (Radna i Stalna), Magistrala (Komunikacija izmedju komponenti), Modem (Komunikacija sa centralnom stanicom)
2. **Specijalizovane komponente:** Ulazno/izlazni analogno/digitalni moduli (prikupljanje podataka sa senzora i upravljanje uredjajima), signalna izolacija i zastita (sprecavanje ostecenja sistema od

prejakih signala i gresaka usled kvarova, 4-20mA)

Komunikacija

Komunikacija između RTU-a i centralne stanice može se organizovati na dva osnovna načina:

1. Ciklično Prozivanje:

- **Opis:** Centralna stanica redom inicira komunikaciju sa svakim RTU-om kako bi proverila stanje i eventualne promene.
- **Prednosti:** Jednostavnost implementacije, niži troškovi.
- **Mane:** Potencijalno **kasnjenje u prenosu kritičnih informacija**, nepotrebna komunikacija kada nema promena.

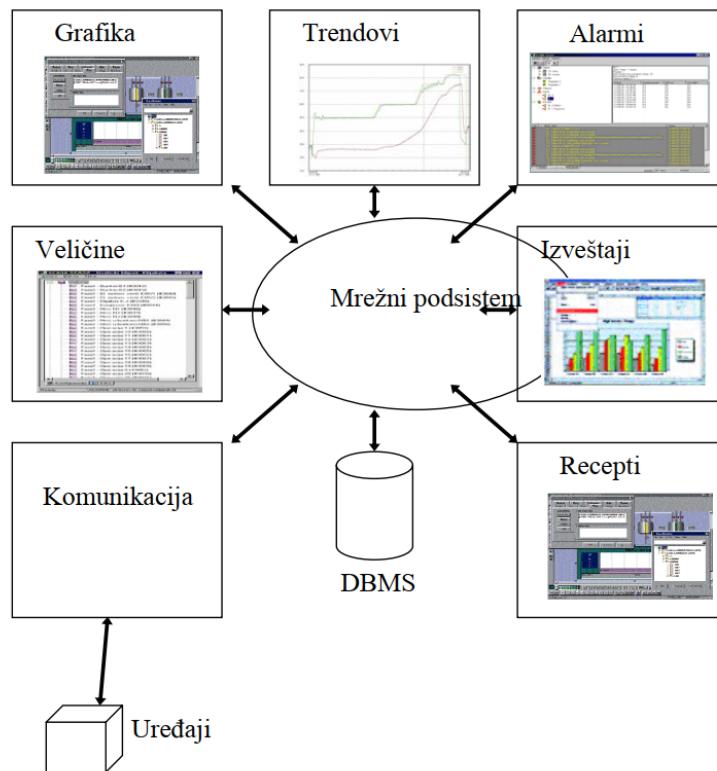
2. Dojave:

- **Opis:** RTU autonomno inicira komunikaciju sa centralom samo kada dođe do promene unutar njegovih parametara.
- **Prednosti:** Efikasnost komunikacije, smanjene kasnjenje i nepotrebna komunikacija.
- **Mane:** Složenost implementacije, veći troškovi, **rizik od kolizije u komunikaciji pri istovremenim promenama.**

Izbor između cikličnog prozivanja i dojava zavisi od broja uređaja u sistemu i potrebe za real time informacijama. Za manje sisteme često se koristi ciklično prozivanje zbog nižih troškova, dok su dojave pogodnije za veće, distribuirane sisteme gde je brza reakcija na promene kritična.

Za validaciju tačnosti podataka prikupljenih sa RTU-a, preporučuje se koristiti više uređaja za merenje istog parametra. Ako dva od tri uređaja daju konzistentne podatke, treći koji odstupa može se smatrati neispravnim.

6. Osnovni zadaci (podsistemi) SCADA sistema



1. Akvizicija (definisanje veličina)

Podsistem za **definisanje veličina** u kome se definisu veličine i njihove osobine kao što su **gornja i donja granica vrednosti veličine, jedinica mere (mA, mV...), vreme očitavanja, inženjerska jedinica (s,Kg,C...) itd.** Ulazne veličine predstavljaju vrednosti izmerenih fizičkih veličina iz procesa, a izlazne veličine vrednosti koje se šalju ka upravljačkim uređajima. Često se mogu definisati i **memorijske veličine** (koje služe za proračune) i **sistemske veličine** koje su specifične za upotrebljeni program.

2. Alarmi

Podsistem za alarne služi za definisanje i prikaz alarmnih stanja u sistemu. Alarmna stanja mogu predstavljati **nedozvoljenu ili kritičnu vrednost veličine** kao i **nedozvoljenu akciju ili komandu operatera**. Svaki alarm ima svoje osobine kao što su **nivo ozbiljnosti alarma, mesto nastanka, kategorija, poruka** koja se vezuje za alarm i slično. Podsistem za alarne omogućuje promenu stanja alarma putem operacije **potvrde i brisanja**.

3. Trendovi

Prikazuju se **poslednje promene** vrednosti veličina (trendovi u realnom vremenu) i **istorijat** promene vrednosti veličina u toku dužeg vremenskog perioda (**histogrami**). Dobro osmišljeni podsistemi za prikaz trendova omogućuju i uporedni prikaz više veličina kao i arhiviranje dijagrama.

4. Komandovanje (recepti)

Omogućuje zadavanje više vrednosti veličina kao željene promene vrednosti veličina u vremenu. Koristi se kada unutar sistema postoje prelazi izmedju stanja kada je potrebno menjati veliki broj veličina u tom sistemu, da ne bismo menjali rucno veličine svaki put, uvodimo neka predefinisanja stanja u kojima se sistem nalazi pri promenama izmedju stanja sistema. Na primer, ako inženjer želi

da pređe sa jednog načina rada opreme na drugi, on može primeniti odgovarajući recept koji automatski podešava sve potrebne parametre opreme kako bi se osigurao taj prelaz. Ovo izbegava ručno podešavanje svakog parametra i smanjuje mogućnost grešaka.

5. Izvestaji

Formiraju se izveštaji o promenama vrednosti veličina, alarmima, akcijama operatera i ostalim aspektima rada postrojenja.

6. Prikaz Stanja (Grafika)

Prikazuje stanje postrojenja u obliku koji je najpregledniji za čoveka (operatera) kako bi on mogao pravovremeno odreagovati na promenu stanja sistema. Osnovna ideja je da se letimičnim pogledom na ekran uoče nepravilnosti u radu postrojenja, da bi se brzo reagovalo i sprečilo neželjeno ponašanje. Vrednosti veličina se najčešće prikazuju u obliku brojeva ili "dinamičkih slika", čime se olakšava uočavanje promena na slici. Pored prikaza stanja sistema grafički podsistem treba da omogući i izvršavanje neke akcije od strane operatera. Na primer klikom miša na neki objekat može se pokrenuti izvršavanje nekog ranije definisanog makroa ili skripta. U većini dostupnih sistema omogućeno je pisanje makroa u VBA (Visual Basic for Application) programskom jeziku koji se odlikuje jednostavnom sintaksom.

7. Komunikacioni Podsistemi (Uredjaji)

Omogućuje povezivanje SCADA sistema sa fizičkim uređajima koji vrše neposredan nadzor i upravljanje (PLC). Najčešće se ovo povezivanje vrši preko drajvera koji su razvijeni od strane proizvođača merne i upravljačke opreme. U SCADA softverskim sistemima nezavisnih proizvođača postoji velika paleta drajvera za opremu različitih proizvođača, dok je kod proizvođača SCADA softvera i upravljačke opreme akcenat stavlja na dajvere za sopstvene uređaje.

8. Skladistenje

Podsistemi za pristup bazama podataka (DBMS - Data Base Management System) omogućuju trajno čuvanje i pregled podataka u relacionim bazama podataka. Ranija rešenja su beležila podatke u datoteke u nestandardnom obliku. Novija rešenja koriste neki od standardnih načina arhiviranja podataka koji omogućuju korisniku lak pristup podacima kao i pristup podacima iz drugih softverskih sistema. Na Microsoft Windows operativnim sistemima često se koristi ODBC (Open Database Connectivity) i nešto savremenija ADO (ActiveX Data Object) tehnologija. Upotreba ovih tehnologija omogućuje lakšu pretragu podataka kao i formiranje izveštaja pomoću SQL (Structured Query Language) jezika.

9. Mrežni Podsistemi

Omogućuje povezivanje ostalih podsistema SCADA programa. Brzina rada celog sistema u mnogome zavisi od programske rešenja ovog sistema. Obično se rešenja oslanjaju na operativni sistem i koriste poznate protokole (transportnog nivoa). Jedna grupa rešenja koristi DDE (Dynamic Data Exchange) tehnologiju koja je jednostavna za korištenje i pokazuje dobre performanse u sistemima u kojima se prenosi mala količina podataka. U složenijim sistemima često je koriste sistemi distribuiranih objekata koji se zanivaju na DCOM i CORBA tehnologijama.



Kaze dragan ne treba 7. i 8. We believe him 🙏

9. Distribuirani računarski sistem (definicija, primeri primene)

Distribuirani računarski sistem (DS) predstavlja skup računara koji su međusobno povezani preko komunikacione mreže. Ovi računari zajedno rade na rešavanju određenog problema ili izvršavanju zadataka, tako što deluju kao jedinstvena celina za krajnjeg korisnika. Zahvaljujući naprednom softveru, distribuirani sistemi integrišu različite aplikacije koje se izvršavaju na različitim računarima u jedan koherentan sistem. Za razliku od centralizovanih sistema, gde je glavni fokus na jednom glavnom računaru, distribuirani sistemi koriste više računara kako bi poboljšali performanse, pouzdanost i skalabilnost.

Primeri

- **Sistem Mobilne Telefonije:** Mobilni telefoni koriste distribuirane sisteme za pružanje usluga kao što su pozivi, slanje poruka i internet konekcije. Ovi sistemi omogućavaju da milioni korisnika širom sveta istovremeno koriste različite usluge.
- **GPS (Globalni Pozicioni Sistem):** Koristi mrežu satelita koji su međusobno povezani i distribuirani širom sveta. Ovi sateliti šalju precizne informacije o lokaciji koje koriste uređaji na Zemlji za navigaciju.
- **World Wide Web:** WWW predstavlja model distribuiranih dokumenata gde su resursi poput web stranica hostovani na različitim serverima širom sveta, omogućavajući korisnicima pristup informacijama nezavisno od njihove lokacije.
- **Sistem Elektronskog Plaćanja:** Sistemi za elektronsko plaćanje, kao što su online bankarstvo i digitalne novčanike, koriste distribuirane sisteme za obradu transakcija u realnom vremenu, omogućavajući sigurne i brze finansijske transakcije.
- **SCADA (Supervisory Control And Data Acquisition) Sistemi:** SCADA sistemi su primeri distribuiranih kontrolnih sistema koji se koriste u industrijskim operacijama za nadzor, upravljanje i prikupljanje podataka sa različitih lokacija u fabrici ili postrojenju.

Razlozi Pojave

- **Brz Razvoj Računarske Tehnologije:** Poboljšanja u hardveru omogućila su izgradnju brzih i pouzdnih distribuiranih sistema.
- **Razvoj Računarskih Mreža:** Brzine mreža od LAN-a (do 10 Gbps) do WAN-a (do 1 Gbps) omogućavaju efikasnu komunikaciju između računara u distribuiranim sistemima.
- **Neophodnost Distribuirane Obrane Informacija:** Složene aplikacije zahtevaju distribuiranu obradu kako bi efikasno upravljale velikim količinama podataka i korisničkim zahtevima.
- **Složenost i Opsežni Zahtevi Korisnika:** Savremene aplikacije zahtevaju kompleksne i brzo skalabilne sisteme koji mogu efikasno upravljati povećanim zahtevima korisnika.

10. Osobine Distribuiranog Sistema

Distribuirani sistemi nude značajne prednosti u pogledu **skalabilnosti, fleksibilnosti i efikasnosti obrade**, pogodni su za složene i velike aplikacije. Međutim, njihova implementacija i upravljanje zahtevaju pažljivo razmatranje zbog inherentne složenosti, potencijalno smanjenih performansi za određene zadatke, i izazova u održavanju sigurnosti. Evaluacija između distribuiranih i centralizovanih sistema treba biti zasnovana na specifičnim potrebama i kapacitetima organizacije, a ne samo na brzini izvršavanja.

Pozitivne Osobine

- **Transparentnost:**
 - **Abstrakcija i Generičnost:** Distribuirani sistemi koriste **middleware**, odnosno sloj koji omogućava aplikacijama da **interaguju sa sistemom na ujednačen način**, nezavisno od hardverskih razlika i složenosti komunikacije. Ovo korisnicima omogućava da koriste sistem **bez potrebe za razumevanjem kompleksnih detalja** kako sistem funkcioniše. Na primer, u distribuiranom sistemu za skladištenje podataka, korisnici jednostavno šalju zahtev za čuvanje podataka ili za njihovo dohvatanje, a ne moraju da znaju na kojem tačno serveru se ti podaci nalaze, kako se upravlja redundancijom ili kako se balansira opterećenje između servera. **Generičnost** znači da se isti interfejs i funkcionalnosti sistema pružaju nezavisno od konkretnih tehničkih implementacija i konfiguracija koje se mogu razlikovati u pozadini
 - **Konzistentnost:** Korisnici i aplikacije se ponašaju u skladu sa **definisanim pravilima korišćenja**, što omogućava dosledno iskustvo bez obzira na to na kojem se računaru ili mreži nalaze.
- **Laka Proširivost:**
 - Distribuirani sistemi su dizajnirani tako da se lako mogu dodavati ili modifikovati kako hardverske tako i softverske komponente **bez prekida rada ili potrebe za velikim rekonfiguracijama**.
- **Podrška za Heterogene Računare i Mreže:**
 - Zahvaljujući slojevitoj organizaciji softvera, distribuirani sistemi mogu efikasno da funkcionišu čak i na heterogenoj mreži **različitih vrsta računara i operativnih sistema**.

Negativne Osobine (U odnosu na centralizovane)

- **Složenost Softvera:**
 - Sinhronizacija među računarima unutar sistema često zahteva složene algoritme kako bi se osigurala doslednost podataka i operacija. **Softver je kompleksniji za razvijanje, konfigurisanje i održavanje**.
- **Smanjene Performanse za Neke Zadatke:**
 - Iako distribuirani sistemi mogu efikasno obrađivati velike i složene zadatke, performanse pojedinih operacija mogu biti manje efikasne zbog **vremena potrebnog za komunikaciju između računara**.
- **Smanjena Sigurnost:**
 - Povećani broj tačaka u mreži može uvesti nove rizike za sigurnost, posebno ako sistem nije adekvatno zaštićen.

- **Teorijska Ograničenja:**
 - Postoje teorijski problemi koje je praktično nemoguće savršeno rešiti u distribuiranim sistemima, kao što su precizna sinhronizacija u velikim sistemima i problemi vezani za kašnjenje i opterećenje mreže.

Middleware (Nije deo pitanja)

Middleware predstavlja sloj softvera koji sedi između aplikacija koje koriste distribuirani sistem i same mreže računara koji čine taj sistem. Middleware tako omogućava usklađivanje komunikacije između različitih komponenti sistema, štiteći aplikacije od složenosti mrežnih operacija, različitih hardverskih konfiguracija i specifičnih implementacija softvera.

Kada koristite VPS, vaš server je jedan od mnogih koji se izvršavaju na istom fizičkom hardware-u. Međutim, softver za virtualizaciju (middleware) omogućava svakom VPS-u da funkcioniše kao nezavisni server. Korisnici upravljaju svojim VPS-om kao da je to zaseban fizički server, iako se hardverski resursi dele između više korisnika. Middleware upravlja resursima i izolacijom, omogućavajući svakom korisniku generički pristup tipičnim resursima servera (CPU, memorija, prostor za skladištenje, itd.) bez uvida u kompleksnost deljenog fizičkog sistema.

11. Ciljevi Distribuiranog Sistema

Distribuirani sistemi su dizajnirani da omoguće efikasno upravljanje resursima, visok nivo transparentnosti, laku integraciju i skaliranje, uz osiguranje visoke sigurnosti. Ovi ciljevi omogućavaju distribuiranim sistemima da podrže složene aplikacije i velike korisničke baze u dinamičnim i zahtevnim operativnim okruženjima.

Povezivanje Resursa sa Korisnicima

Distribuirani sistemi omogućavaju korisnicima efikasno deljenje i korišćenje različitih resursa, kao što su:

- **Deljenje Uređaja:** Omogućava više korisnika da dele hardverske resurse poput štampača, skenera, i diskova.
- **Razmena Datoteka:** Podrška za jednostavnu razmenu dokumenata, emailova, audio i video zapisa.
- **Rad na Daljinu:** Podržava aplikacije za telekonferencije, elektronsku trgovinu, i druge oblike udaljenog rada.

Ova karakteristika je vitalna za optimizaciju resursa i efikasnost rada u savremenim poslovnim i tehnološkim okruženjima.

Transparentnost

Jedan od glavnih ciljeva distribuiranog sistema je da korisnicima pruži iskustvo rada na jednom koherentnom sistemu, čime se sakriva kompleksnost unutrašnjeg funkcionisanja sistema.

Korisnicima deluje kao da rade na jednom racunaru:

- **Sakrivanje Kompleksnosti:** Korisnicima nije potrebno da razumeju detalje rada mreže i distribucije podataka.

- **Konzistentno Korisničko Iskustvo:** Bez obzira na to gde se resursi fizički nalaze, korisnici ih koriste na isti način.

Otvorenost

Otvorenost u distribuiranim sistemima omogućava jednostavno dodavanje, menjanje ili uklanjanje komponenti bez ugrožavanja celokupnog sistema:

- **Standardizacija:** Servisi su dizajnirani po standardnim pravilima, što olakšava integraciju i komunikaciju između različitih proizvođača i sistema. Servisi se obično specificuju preko interfejsa (mesto gde 2 uredjaja komuniciraju). Mogucnost da kroz dobro definisan interfejs imamo sistem gde odredjenu komponentu mozemo da zamenimo njenim ekvivalentom. Dobro definisan interfejs je:
 - Kompletan - Specifirano je sve sto treba implementirati
 - Neutralan - Implementacioni detalji nisu spolja vidljivi
 - Interoperabilnost - Delovi sistema raznih izvodjaca mogu da rade zajedno i komuniciraju preko interfejsa
 - Portabilnost - Aplikacija razvijena za sistem A se može izvrsavati (bez modifikacija) u sistemu B (koji ima iste interfejse kao i A)
- **Fleksibilnost:** Sistem može lako da se prilagođava i proširuje sa novim funkcionalnostima.

Skalabilnost

Skalabilan je u suprotnosti sa **centralizovan**

- Centralizovan servis – izvršava se na samo jednom serveru
- Centralizovani podaci – nalaze se samo na jednom mestu
- Centralizovan algoritam – odluka se donosi samo na osnovu kompletne informacije

Distribuirani sistemi su dizajnirani da efikasno skaliraju u skladu sa rastućim zahtevima:

1. Prosirenje kapaciteta: dodavanje novih korisnika i resursa – **opterećenje raste**
 - **Horizontalno Skaliranje:** Dodavanje više istih ili sličnih komponenata u sistem za povećanje kapaciteta i otpornosti na greške. Mana je **kompleksnost**.
 - **Vertikalno Skaliranje:** Povećavanje kapaciteta postojećih komponenata, poput dodavanja više memorije ili jačih procesora. **Jednostavno** se sprovodi (obično ne zahteva programsku promenu) ali ograniceno brojem komponenti.
2. Geografska distribuiranost: proširujem mesta na kojima pružam servise; geografsko proširenje sistema (pristup sa udaljenih mesta) –**kašnjenja i manja pouzdanost veza**

Sigurnost

Bezbednost sistema je ključna u svim aspektima distribuiranih sistema, posebno zbog njihove otvorenosti i složenosti. Ovo uključuje zaštitu podataka, osiguranje integriteta informacija, kao i zaštitu od neautorizovanog pristupa.

12. Tipovi distribuiranih sistema (Klaster, Grid Computing)

Distribuirani sistemi se mogu klasifikovati u dva osnovna tipa na osnovu njihove organizacije, kapaciteta i geografske distribucije: klaster i grid computing. Ovi sistemi imaju različite karakteristike i primene u različitim tehnološkim i poslovnim okruženjima.

Klaster Sistemi

Klaster sistemi su grupa računara, često identičnih, koji se nalaze na istoj geografskoj lokaciji i povezani su u mrežu. Svi računari u klasteru tipično pokreću isti operativni sistem. Klasteri se koriste za postizanje većih performansi kroz paralelno procesiranje zadataka. Svaki računar u klasteru može raditi na delu većeg problema, čime se postiže znatno brže procesiranje.

Primenjuju se za intenzivnu obradu podataka; Naučna istraživanja koja zahtevaju velike računske snage. Podrška za aplikacije sa visokim zahtevima za procesiranje, kao što su velike baze podataka ili aplikacije za obradu slika.

Prednosti su visoka performansa i pouzdanost, lakše upravljanje jer svi delovi klastera dele isti sistem i politike, visoka kontrola nad resursima i aplikacijama.

Grid Computing

Grid computing uključuje računare koji su geografski raspršeni i često nehomogeni u pogledu hardvera, operativnih sistema i mrežne infrastrukture. Računari unutar grid sistema mogu pripadati različitim organizacijama i mogu imati različite administratore. Grid se oslanja na softver za virtualizaciju i srednje slojeve za koordinaciju složenih zadataka preko različitih računarskih sistema.

Primenjuju se za procesiranje velikih količina podataka koje zahtevaju računarske resurse širom sveta; Naučna istraživanja koja zahtevaju pristup dislociranim računarskim resursima, kao što su projektovanje novih lekova ili klimatska istraživanja, projekti koji zahtevaju kombinovanje resursa iz više izvora, kao što su veliki eksperimenti u fizici.

Prednosti su ekstremna skalabilnost i fleksibilnost; Mogućnost korišćenja izuzetno velikih resursa koji nisu mogući unutar jednog klastera ili organizacije; Resursi se mogu dinamički dodavati ili oduzimati prema potrebi.

Razlike

- **Kontrola:** Klasteri su više kontrolisani i predvidivi jer su konfigurisani na istoj lokaciji sa istim operativnim sistemom. Grid sistemi su manje predvidivi zbog heterogenosti i geografske disperzije.
- **Administracija:** Klasteri su lakši za administraciju zbog homogenosti sistema, dok grid sistemi zahtevaju složeniju koordinaciju i upravljanje.
- **Zahtevi za performanse:** Klasteri su optimalni za zadatke koji zahtevaju intenzivno računanje u realnom vremenu. Grid sistemi su bolji za zadatke koji mogu tolerisati određeno kašnjenje i zahtevaju izuzetno velike računske kapacitete distribuirane na širem geografskom području.

13. 5 Koraka Omron i IFIX SCADA Sistema

Intellution IFIX je **SCADA softver** koji razvija kompanija GE Digital. Ovaj softver omogućava nadgledanje, kontrolu i analizu rada industrijskih procesa i sistema kroz stvaranje naprednih grafičkih korisničkih interfejsa (HMI) i baza podataka za praćenje tagova. IFIX omogućava korisnicima da **razviju skripte** i logiku kontrola u svojoj SCADA platformi, što znači da su u stanju da implementiraju složene kontrolne algoritme direktno kroz SCADA sistem.

Sa druge strane, **OMRON CX-Supervisor** je **SCADA softver** razvijen od strane OMRON-a. CX-Supervisor omogućava korisnicima da nadgledaju i kontrolišu industrijske procese putem naprednih HMI dizajna i skriptovanja logike. Ovaj softver se koristi zajedno sa OMRON PLC-ovima za integriranu kontrolu i nadzor sistema.

OMRON takođe nudi **CX-Programmer** za programiranje logike PLC-a i **CX-Designer** za dizajniranje HMI ekrana. PLC-ovi (Programmable Logic Controllers) su fizički uređaji koji upravljaju industrijskom automatizacijom na terenu. OMRON PLC-ovi procesiraju programsku logiku unutar strujno-orientisanih instalacija, a koriste se za prikupljanje podataka sa senzora, upravljanje motorima, aktuatorima i drugim komponentama unutar proizvodnog sistema.

U osnovi, **IFIX** je više usmeren na **SCADA sistemske aspekte**, poput vizualizacije podataka i upravljanja operacijama preko računarskog sistema, dok je **OMRON CX-Supervisor** fokusiran na **integriranu SCADA i PLC kontrolu**, omogućavajući korisnicima da dizajniraju HMI i razvijaju kontrolnu logiku specifičnu za OMRON-ove PLC-ove. Međutim, postoji nekoliko osnovnih koraka koji su uobičajeni za proces postavljanja i upravljanja ovim automatskim sistemima na obe platforme.

1. Konfiguracija i Postavljanje

- **IFIX:** Uključuje postavljanje SCADA okruženja, definisanje arhitekture sistema, konfigurisanje čvorova i postavljanje komunikacionih protokola.
- **OMRON CX-Supervisor:** Uključuje postavljanje SCADA okruženja, definisanje arhitekture sistema, konfigurisanje čvorova i postavljanje komunikacionih protokola specifičnih za OMRON PLC uređaje.

2. Kreiranje Baze Tagova

- **IFIX:** Kreiranje baze tagova koji predstavljaju različite veličine prikupljene sa senzora, aktuatora i kontrolnih tačaka unutar SCADA sistema.
- **OMRON CX-Supervisor:** Kreiranje baze tagova koji predstavljaju različite veličine prikupljene sa senzora, aktuatora i kontrolnih tačaka unutar SCADA sistema, sa direktnim mapiranjem na OMRON PLC promenljive.

3. Programiranje i Razvoj Logike

- **IFIX:** Razvijanje skripti ili logike koristeći jezike za skriptovanje SCADA sistema za implementaciju funkcija kontrole i nadgledanja.
- **OMRON CX-Supervisor:** Razvijanje skripti koristeći VBScript ili slične jezike za implementaciju kontrolne logike i nadgledanje, uz integraciju sa PLC programima.

4. Dizajn HMI (Human-Machine Interface)

- **IFIX:** Dizajniranje i konfigurisanje grafičkih korisničkih interfejsa za operatere kako bi nadgledali i kontrolisali proces, uključujući kreiranje ekrana, animacija i korisničkih kontrola.
- **OMRON CX-Supervisor:** Dizajniranje i konfigurisanje grafičkih korisničkih interfejsa koristeći CX-Supervisor softver, uključujući kreiranje ekrana, animacija i kontrola specifičnih za OMRON PLC-ove.

5. Testiranja i Validacija

- **IFIX:** Sprovodjenje temeljnih testiranja SCADA sistema, uključujući simulaciju uslova procesa, verifikaciju komunikacije i validaciju kontrolne logike.
- **OMRON CX-Supervisor:** Sprovodjenje temeljnih testiranja SCADA sistema, uključujući simulaciju uslova procesa, verifikaciju komunikacije sa OMRON PLC-ovima i validaciju kontrolne logike.

Random 💀 Nekoherentne 💀 Beleske

Vrv nije deo pitanja (sigurno)

- **Pneumatski Ventil:** U IFIX SCADA sistemima, ventil se koristi za kontrolu protoka tečnosti ili gasa preko pritiska koji deluje na membranu. IFIX može koristiti tagove za nadzor pritiska i otvorenosti ventila, što omogućava precizno regulisanje protoka.
- **Diferencijalni Merač Pritiska:** Ovaj instrument se koristi za merenje razlike u pritisku pre i posle procesnog elementa, što omogućava određivanje brzine fluida. IFIX prati ove vrednosti preko tagova i koristi ih za upravljanje procesima.
- **OPC Standard:** IFIX koristi OPC (OLE for Process Control) standard za komunikaciju sa PLC uređajima i drugim krajnjim elemenatima, što omogućava laku integraciju i interoperabilnost.
- **Alarmi i Vizualizacija:** IFIX omogućava dinamično prikazivanje aktivnih i nedavnih alarma, jasno ističući neobrađene alarme u crveno. Svaki alarm može biti detaljno definisan sa imenom, prioritetom, veličinom koju prati, i logičkim izrazom za aktivaciju.
- **Servo Ventil:** Ovaj ventil, koji se često koristi u OMRON PLC sistemima, omogućava preciznu kontrolu protoka. Integracija sa OMRON PLC-ovima omogućava izvođenje zadatih operacija na fluidima kontrolisanih servo ventilima.
- **Grafički i Brojčani Prikazi:** OMRON SCADA sistem omogućava dizajniranje HMI (Human-Machine Interface) interfejsa koji uključuju i grafički i numerički prikaz svakog elementa procesa, unapređujući preciznost i lakoću nadzora.
- **Definisanje Alarmnih Parametara:** OMRON SCADA sistem omogućava korisnicima da precizno definiraju uslove koji će pokrenuti alarm, kao i da implementiraju funkcije kao što su "deadband", koji ignoriše manje varijacije u veličinama, i "rate of change", koji detektuje i upozorava na nagla i neobična odstupanja.
- **Integracija i Scripting:** OMRON sistemi omogućavaju korišćenje Visual Basic i drugih skriptnih jezika za integraciju SCADA sistema sa drugim aplikacijama, unapređujući fleksibilnost i proširivost sistema.
- Animacije i vizuelni signali poput jarkih boja, treptanja i brzih promena takođe se koriste za privlačenje pažnje na kritične promene unutar procesa, što omogućava operatorima da efikasno reaguju pomoću perifernog vida.

- Takođe, programski jezici koji se koriste u SCADA sistemima su interpretirani, što znači da se izvršavaju "on the fly" i omogućavaju dinamičnu interakciju i brzu adaptaciju na promene u procesnim parametrima. Ova fleksibilnost je ključna za industrijske aplikacije gde se uslovi često mijenjaju.

14. OPC (+WinCC)

OPC - Object Linking and Embedding For Process Control

OPC Fondacija, formirana sa idejom da unapredi interoperabilnost među industrijskim automatizacijskim uređajima, predstavlja organizaciju koja razvija i održava OPC standard. Osnovna ideja ove fondacije je da stvori ujednačene i standardizovane komunikacione protokole, koji omogućavaju pouzdanu i efikasnu razmenu podataka između različitih kontrolnih sistema i uređaja, unapređujući integraciju i automatizaciju industrijskih procesa.

| **idemo po tu desetku bre takoe!** 😊

Komunikacioni standard koji omogućava pouzdano **povezivanje različitih industrijskih uređaja** i kontrolnih sistema, kao što su SCADA sistemi, PLC uređaji i druge aplikacije. Standardizacija preko OPC omogućava jednostavniju integraciju i komunikaciju među komponentama različitih proizvođača, značajno smanjujući složenost integracije i održavanje sistema.

U praksi nije redak slučaj da su pojedina postrojenja u okviru istog preduzeća geografski udaljena. Tako se može dogoditi da se nadzorno-upravljački sistemi za upravljanje pojedinim postrojenjima razvijaju pomoću komponenti različitih proizvođača.

OPC fondacija - uvođenje standardizovanih međuprocesnih komunikacija u industriji. Ideja je da se sva **komunikacija između SCADA sistema i uređaja uniformiše**. Članovi su svi vodeći proizvođači Microsoft, Siemens, Omron, NI, Festo...

OPC = OLE for Process Control

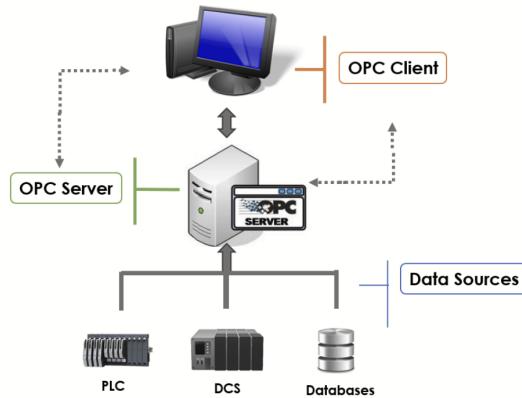
OLE = Object Linking and Embedding

OPC standard je **skup interfejsa, metoda sa svim parametrima i pravila** koja moraju biti postovana od strane svih proizvodjaca. OPC fondacija nije implementirala metode vec svaki proizvodjac, a **fondacija samo zadaje parametre**.

Mesta primene ovih standarda uključuju:

- **Povezivanje komponenti distribuiranog upravljačkog sistema.** OPC omogućava komunikaciju između PLC-a u jednoj fabrici i SCADA sistema u drugoj lokaciji, što omogućava centralizovano upravljanje.
- **Pristup podacima postojećeg upravljačkog sistema** – Napravljen je **OPC softverski omotač**. Implementacija OPC softverskog omotača omogućava jednostavno povezivanje starog DCS (Distributed Control System) sa novim SCADA sistemom bez potrebe za kompletnom zamenom hardvera.
- **Pristup podacima upravljačkih uređaja** (mernoakvizicionih uređaja) - OPC server omogućava prikupljanje podataka o temperaturi i pritisku direktno sa senzora postavljenih na mašinama, što omogućava realno vreme nadzora i analize.

Unutar OPC fondacije postoje 3 grupe objekata:



1. **Server** - predstavlja najvisi objekat u hijerarhiji
2. **Item** - predstavljaju fizicke velicine kojima se upravlja
3. **Grupa** - nema fizicku reprezentaciju, predstavlja skup item-a, omogucuje optimalan pristup velicinama. Organizuje velicine (items) po lokaciji, vremenu, itd.

Specifikacije unutar OPC standarda

Sve specifikacije su klijent/server orientisane. Klijent je taj koji inicira samu komunikaciju, server prihvata njegov zahtev, obradjuje ga i prosledjuje odgovor nazad klijentu. Vazno je napomenuti da je komunikacija izmedju klijenta i servera moze biti **sinhrona** i **asinhrona**.

- **Sinhrona komunikacija** je blokirajuca, sto znači da klijent ostaje blokiran, sve dok mu server ne odgovori. Prednost sinhronne komunikacije jeste u tome što smo na taj nacin sigurni da će klijent uvek dobiti odgovor, dok mu je manja sama blokiranost klijenta. *Sinhroni pristup je namenjen jednostavnim klijentima koji pristupaju malom broju veličina.*
- **Asinhrona komunikacija** nije blokirajuca, što je i sama prednost ovog nacina komunikacije, međutim prilikom ovakvog vida komunikacije klijent nikad ne zna kada će mu stići odgovor od servera. Asinhroni pristup minimizuje upotrebu CPU i mreže.

Dve najznačajnije specifikacije unutar OPC standarda su:

- **DA specifikacija (Data Access)** - pristup samim velicinama
- **HDA (History Data Access)** - pristupa velicinama na istorijskom nivou.

DA specifikacija

Data Access specifikacija služi za brz pristup trenutnim vrednostima velicina. Jedan klijent može da komunicira sa više servera, a jedan server može da opslužuje više klijenata (N:N). Postoje 3 osnovna atributa koja blize određuju samu vrednost velicine:

1. **Vrednost** - npr. temperatura je 44C
2. **Timestamp** (vreme kad se ocita vrednost) - Nije trenutak kada se vrednost promeni!
3. **Kvalitet** - može biti NULL, questionable, bad... Sama specifikacija podržava i sinhronu i asinhronu komunikaciju.
4. **Dodatni atributi** poput: granice vrednosti, granice alarma...

Predviđen je mehanizam **dojave samo promenjenih vrednosti**.

Deadband - Upotreba **mrtve zone** sprečava prenos malih promena kako se ne bi dojavljivao šum. Mrtva zona može da se upotrebni samo u slučajevima kojima se upravlja većinski u stacionarnom stanju.

Update rate - Upotreba **odložene dojave** sprečava česte dojave. Omogućava klijentu da definiše minimalan period uzmeđu dojava sa server.

Adresni prostor je skup svih veličina u sistemu. **Ravan adresni prostor** je onaj u kom su sve veličine iste, pripadaju istoj grupi. A **hijerarhijski adresni prostor** je organizacija itema po grupama.

HDA specifikacija

History Data Access specifikacija služi za čuvanje istorijskih veličina i njihov prikaz, tj. očitavanje. Playback omogućava reproduciju toka podataka u određenom vremenskom opsegu u tačnom redosledu kako su se pojavljivali. Pomocu Playback interfejsa omoguceno je da u ponovljenom, realnom vremenu vidimo promene vrednosti podataka, alarne i slično. Klijent može da bira na koliko sekundi zeli citanje podataka (Scan ciklus).

Opisuju interfejse bitne za:

- skladištenje vrednosti veličina
- očitavanje ranije prikupljenih/skladištenih vrednosti

Specifikacije predviđaju postojanje:

- jednostavnijih servera koji bi implementirali samo osnovne funkcionalnosti istorijskog servera
- složenih servera kod kojih se vrše filtracije podataka, statistička obrada, kompresija podataka i sl.

I ovde je predviđen mehanizam **dodatake** kao najefikasniji način prenosa podataka između klijenata i servera. Sinhrona komunikacija blokira dok se zahtev ne ispunii.

Postoje veličine i imaju osobine (kao u OPC Data Access).

1. Item (tag) je osnovna komponenta, jedna fizička veličina.
2. Opseg željene vrednosti
3. Raw value - sirov, očitan podatak
4. Agregiran podatak - avg, min, max ... Ove vrednosti dozvoljavaju pretragu npr od min do max
5. Anotacije – korisnik može komentare dodeliti veličinama (uz vremensku značku) i očitati ih
6. Interpolirani podaci – vrednosti veličina dobijene linearnom interpolacijom
7. Modifikovane vrednosti – promenjene nakon skladištenja u server.

Nema grupa u HDA. Moramo znati tačno šta tražimo kad pristupamo istoriji.

WinCC

WinCC (Windows Control Center) je SCADA (Supervisory Control and Data Acquisition) sistem razvijen od strane kompanije **Siemens**. WinCC mreža u kontekstu SCADA-e odnosi se na arhitekturu koja omogućava komunikaciju i integraciju različitih komponenti sistema za nadzor i kontrolu industrijskih procesa. Evo kratkog objašnjenja ključnih aspekata:

- Arhitektura:** WinCC mreža je zasnovana na **distribuiranoj arhitekturi** koja omogućava povezivanje više računara i uređaja. Ova mreža može uključivati centralni server, klijentske računare, PLC-ove (Programabilni Logički Kontroleri), HMI (Human-Machine Interface) uređaje i druge mrežne komponente.
- Komunikacija:** WinCC koristi različite komunikacione protokole kao što su OPC i drugi za prenos podataka između servera, PLC-ova i HMI uređaja. To omogućava centralizovani nadzor i kontrolu nad različitim delovima proizvodnog procesa.
- Skalabilnost i Fleksibilnost:** WinCC mreža je skalabilna, što znači da se može lako prilagoditi promenama u obimu i složenosti proizvodnog procesa. Dodavanje novih uređaja, senzora i kontrolnih jedinica može se obaviti bez značajnih promena u postojećoj infrastrukturi.

WinCC može da poveže servere različitih proizvođača i omogući razmenu podataka.

- Sigurnost:** Sigurnosne funkcije u WinCC mreži uključuju **autentifikaciju korisnika, kontrolu pristupa i enkripciju podataka** kako bi se osigurala zaštita od neovlašćenog pristupa i manipulacije.

WinCC mreža omogućava integrисано и ефикасно управљање индустријским процесима кроз pouzдану комуникацију, real-time praćenje и sveобухватну vizualizацију података, што је ključно за modernu industrijsku automatizaciju.

15. Cloud

Definicija Cloud Computinga

Cloud computing predstavlja revolucionarni **način pružanja IT resursa preko interneta**, omogućavajući korisnicima **pristup velikim skladišnim, računskim i mrežnim kapacitetima** bez potrebe za održavanjem vlastite infrastrukture. U osnovi, cloud je operativni sistem za velike Data Centre, omogućavajući **efikasno upravljanje resursima i optimalnu raspodelu radnog opterećenja**.

Operativni sistem za Data Centre: Cloud infrastruktura funkcioniše kao operativni sistem koji управља свим resursима у velikim Data Centrima. То укључује računarsке (compute), skladišне (storage) и мрежне (network) resurse. Pruža okruženje у којем aplikације могу да rade efikasno i bez ometanja.

Upravljanje resursima: Cloud sistemi су dizajnirani да **automatski upravljaju dostupnim resursima**, omogućavajući **optimalnu alokaciju i korišćenje resursa**. Ovo помаже у смањењу трошкова и пovećava efikasnost, jer resursi могу бити динамички прilagođeni потребама корисника.

Životni ciklus aplikacija: Cloud platforme definišu životni ciklus aplikacija, од развоја и testiranja до implementacije и održavanja. Automatski upravljaju različitim fazama životnog ciklusa, чime se smanjuje vreme потребно за implementaciju novih aplikacija i ažuriranja postojećih.

Viši nivo abstrakcije: Jedna od ključnih предности cloud computinga je **pružanje višeg nivoa abstrakcije**. То znači да корисници не moraju да се баве složеним detaljima инфраструктуре, већ могу да се фокусирају на развој и управљање aplikацијама. Cloud platforme **apstraktuju fizičku инфраструктуру**, omogućavajući jednostavnije korišćenje resursa.

Skalabilnost: Cloud sistemi су дизајнирани да се **lako skaliraju**, omogućavajući корисницима да пovećају или смање своје resurse **u skladu sa trenutnim potrebama**. Ovo је posebno корисно за

preduzeća koja imaju promjenjive potrebe za IT resursima, jer omogućava fleksibilnost i brzo prilagođavanje promenama.

Iluzija beskonačnih resursa: Jedna od najprivlačnijih karakteristika cloud computinga je **iluzija beskonačnih resursa**. Korisnici imaju osećaj da im je na raspolaganju neograničena količina resursa, jer cloud platforme omogućavaju dinamičko i neprekidno skaliranje prema potrebama korisnika.

Prednosti Cloud Computinga

Ekonomija velikih brojeva

Veliki dobavljači (vlasnici Cloud-a) imaju značajnu prednost u odnosu na pojedinačne organizacije kada je u pitanju nabavka hardverskih resursa. Zbog obima svojih narudžbina, **veliki cloud provajderi mogu nabaviti** servere, skladišne sisteme i mrežnu **opremu po znatno nižim cenama** nego što bi to mogla bilo koja pojedinačna organizacija. Ova ekonomija obima omogućava cloud provajderima da ponude svoje usluge po konkurentnim cenama, **čime se smanjuju troškovi za krajnje korisnike**.

Pay as you go (Plaćanje po upotrebi)

Plaćanje po upotrebi je model koji omogućava **korisnicima da plaćaju samo za one resurse koje trenutno koriste**. Ako su vam potrebni dodatni resursi, nema potrebe za kupovinom novih servera; umesto toga, jednostavno ih iznajmите od cloud provajdera. Kada vam ti resursi više nisu potrebni, možete ih otkazati i prestati plaćati za njih. Ovaj model pruža visoku fleksibilnost i omogućava optimizaciju troškova, jer **plaćate samo za ono što stvarno koristite**.

Niska početna cena

Niska početna cena je jedna od ključnih prednosti cloud computinga. **Korisnici ne moraju da ulažu u skupu hardversku infrastrukturu** unapred, niti su vezani dugoročnim ugovorima. Ovo omogućava brzi početak rada sa minimalnim inicijalnim troškovima, što je posebno korisno za startape i male firme koje nemaju velike budžete za IT ulaganja.

Visoka dostupnost

Visoka dostupnost znači da su vaše **aplikacije dostupne 24 sata dnevno, 7 dana u nedelji, 365 dana u godini**. Cloud provajderi koriste redundantnu infrastrukturu i napredne tehnologije za **balansiranje opterećenja i otklanjanje kvarova**, što omogućava da vaše aplikacije budu uvek dostupne i operativne. Ovo je ključno za poslovne aplikacije koje zahtevaju konstantnu dostupnost i minimalne prekide u radu.

Nivoi usluga u Cloud-u

Cloud computing omogućava zakup računarskih resursa po potrebi, što znači da plaćate samo za ono što koristite. Ovaj model pruža fleksibilnost i omogućava optimizaciju troškova, jer nema potrebe za velikim inicijalnim ulaganjima u hardver. Resursi koji se mogu zakupiti uključuju procesore (CPU), RAM memoriju, mrežne resurse, i druge ključne komponente IT infrastrukture.

Najveći Cloud Provajderi

Najveći provajderi cloud usluga su:

- **Amazon Web Services (AWS)**

- **Google Cloud Platform (GCP)**
- **Microsoft Azure**

Ovi provajderi nude širok spektar usluga koje omogućavaju korisnicima da prilagode svoje resurse prema potrebama.

Virtuelne Mašine

Da bi cloud computing funkcijonisao, koriste se **virtuelne mašine (VM)** koje simuliraju fizičke računare. Virtuelne mašine omogućavaju korisnicima da instaliraju i koriste operativne sisteme i aplikacije kao da su na fizičkom hardveru. Amazon je jedan od vodećih proizvođača virtuelnih mašina, dok je HP najveći proizvođač hardvera za data centre.

Prednosti Clouda

1. **Plaćaš samo ono što koristiš:** Fleksibilnost u troškovima.
2. **Niska početna cena:** Minimalna ulaganja na početku.
3. **Visoka dostupnost:** Konstantan pristup resursima i aplikacijama.

Data Storage

Data storage u cloudu se sastoji od **velikih skupova hard diskova** koji su dizajnirani da **osiguraju podatke i pruže visoku propusnost**. Ove skladišne jedinice mogu se zakupiti prema potrebi korisnika, omogućavajući im skaliranje skladišnih kapaciteta u skladu sa rastom podataka.

Nivoi Usluga u Cloud Computingu

1. **SaaS (Software as a Service):** Usluga koja **nudi poslovnu logiku direktno krajnjim korisnicima**. Klijenti koriste aplikacije bez potrebe da znaju koji algoritmi i infrastruktura stoje iza njih. Plaćaju samo uslugu koju koriste.
2. **IaaS (Infrastructure as a Service):** Omogućava korisnicima da dobiju **prazan operativni sistem**, pružajući im potpunu **kontrolu nad IT infrastrukturom**. Korisnici mogu **instalirati sopstvene aplikacije i upravljati njima**.
3. **PaaS (Platform as a Service):** Nudi korisnicima platformu koja uključuje **baze podataka i druge razvojne alate**, omogućavajući im da razvijaju, testiraju i distribuiraju aplikacije bez brige o osnovnoj infrastrukturi.
4. **HaaS (Hardware as a Service):** Pruža korisnicima **pristup fizičkom hardveru**, omogućavajući im da koriste resurse kao da su njihovi. Ova usluga je često kombinovana sa ugovorima o pružanju usluga koji definišu uslove korišćenja i održavanja.