

Spremljanje učinkovitosti proizvodnih procesov z obdelavo na robu omrežja

Jure Špeh, Gašper Mušič

Univerza v Ljubljani, Fakulteta za elektrotehniko

E-pošta: jure.speh@student.uni-lj.si

Production process efficiency monitoring with edge computing

Machines and devices nowadays produce large amounts of data that are typically transmitted and stored on remote servers. This can lead to high data transfer and storage costs, delays, as well as security and privacy issues. Data processing at the edge of the network allows the data of sensors and devices to be analyzed and processed in their immediate proximity, and only key information is sent to the cloud for further processing.

The article describes a practical example of monitoring the effectiveness of the production process by calculating the overall equipment effectiveness (OEE) indicators at the edge of the network. The TwinCAT environment runs a production line simulation based on the PackML state model. The TwinCAT device is connected to a Raspberry Pi computer running a program written in Python programming language, which captures data from the production process and calculates OEE indicators in real-time. These are then transferred to the Azure IoT Central web application, where the operator has remote supervisory control over the effectiveness of the process.

1 Uvod

Sodobna industrija proizvaja vedno večje količine podatkov. Ti se navadno iz proizvodnih procesov pretakajo v podatkovne centre, kjer se iz njih pridobiva koristne informacije. Shranjevanje in obdelava velike količine podatkov v oblaku pa ni vedno najbolj smiselna rešitev. Pomisleki se pojavljajo predvsem pri stroških, zasebnosti, zakasnitvah in zanesljivosti. Ponudniki oblačnih rešitev navadno zaračunavajo njihove storitve glede na porabo virov. Nekateri parametri se zaradi varstva osebnih podatkov ne smejo hraniti v podatkovnih centrih. Pri prenosu po omrežju lahko pride do velikih zakasnitev, ki lahko kritično vplivajo na izvedbo storitev. Vse te težave lahko odpravimo z obdelavo in filtriranjem podatkov na robu.

V zadnjem času se obdelava podatkov vse bolj seli na rob omrežja, blizu senzorjem in napravam, ki jih proizvajajo. Tehnologija obdelave podatkov na robu omrežja (ang. edge computing), je oblika decentralizirane obdelave podatkov, ki lahko vključuje napovedne modele in napredne metode strojnega učenja.

V tem članku so predstavljene možnosti in izzivi, ki jih prinaša ta tehnologija. Prikazana je umestitev obdelave podatkov na robu v širši kontekst industrije 4.0, nato pa je predstavljen praktičen primer industrijske aplikacije, kjer najprej z obdelavo na robu izračunavamo kazalnike skupne učinkovitosti proizvodnega procesa (OEE) in jih nato prikazujemo v spletni aplikaciji Microsoft Azure IoT Central.

2 Obdelava podatkov na robu

Oblačne tehnologije so že nekaj časa jedro delovanja številnih organizacij po svetu. Naprave iz interneta stvari se v oblak povezujejo preko internetne povezave, z namenom procesiranja podatkov, ki jih zajemajo. Časovne zakasnitve pri prenosu pa lahko prinesejo velike težave pri delovanju sistemov, zlasti pri aplikacijah, ki delujejo v realnem času. Tak primer so sistemi nadzora zdravja, avtonomna vozila, video nadzor in avtomatsko vodeni proizvodni procesi. S skokovitim porastom naprav IoT in kibernetsko-fizičnih sistemov (CPS)¹ je prišlo do spoznanja, da imajo lahko zakasnitve zaradi procesiranja podatkov v oblaku kritične posledice. To je bil glavni razlog za selitev virov obdelave informacij iz oddaljenih strežnikov na rob omrežja, čim bližje napravam, ki zajemajo telemetrijo in izvajajo nadzor nad realnimi aplikacijami [1]. Robne naprave med drugim omogočajo izvajanje računskih operacij, ki bi sicer tekle v oblaku (razbremenitev strežnikov), dolgoročno in kratkoročno shranjevanje podatkov in razporejanje ukazov (ang. request distribution). Obdelava na robu je smiselna predvsem za srednje oz. nizko zahtevne operacije. Če gremo na nivo senzorjev in naprav, je to mogoče doseči, saj tako omejimo potrebo po sočasni obdelavi velike količine telemetrije naenkrat.

3 Industrijski internet stvari (IIoT)

Termin internet stvari (ang. internet of things) je bil prvič uporabljen leta 1999. Gre za napreden sistem različnih senzorjev, strojev in ostalih naprav, ki lahko komunicirajo med seboj ali s centralnim sistemom v oblaku. Pri komunikaciji naprava-z-napravo (ang. Machine-To-Machine) gre za medsebojno komunikacijo med napravami, ki ne

¹Sistem, kjer je fizični mehanizem nadzorovan oz. upravljan s strani računalniškega algoritma.

potrebujejo posrednika za prenos podatkov. Ta način se pogosto uporablja v sistemih pametnih zgradb. Z uporabnikovega stališča se lahko težava pojavi pri skladnosti naprav različnih proizvajalcev, ki uporabljajo svoje protokole. "Stvari" so v IoT tako proizvajalci kot uporabniki podatkov.

Industrijski internet stvari (IIoT) je še vedno širok pojem, saj zajema skupino zelo različnih industrij, na primer industrijska proizvodnja, rudarstvo, logistika in energetika. Na vseh področjih prinaša napredek v varnosti, optimizaciji stroškov, kakovosti izdelkov in storitev ter produktivnosti.

3.1 Skupna učinkovitost opreme

Skupna učinkovitost opreme je osrednja komponenta metodologije TPM (Total Productive Maintenance), ki izvira iz Japonske. Gre za merilo, ki nam pove stopnjo učinkovitosti proizvodnega procesa v primerjavi s pričakovanimi rezultati [3]. S tem merilom lahko identificiramo, koliko časa je bila oprema v proizvodnji resnično produktivna. Rezultat se meri v odstotkih, kjer 100 % pomeni, da v proizvodnji ni bilo zastojev, med izdelki, ki so bili narejeni v najhitrejšem možnem času, pa so bili samo dobri kosi. Izračun (enačba 1) temelji na treh komponentah:

- Razpoložljivost A (ang. Availability)
- Zmogljivost P (ang. Performance)
- Kakovost Q (ang. Quality)

$$OEE = A \cdot P \cdot Q \cdot 100 \% \quad (1)$$

3.2 PackML

PackML (ang. packaging machine language) je industrijski tehnični standard za vodenje in nadzor pakirnih strojev in linij [4]. Njegov avtor je organizacija OMAC (Organization for Machine Automation and Control) v sodelovanju z združenjem ISA (International Society of Automation) na principu standarda ISA-88 [5, 6]. Glavni namen PackML je izdelati smernice, ki bi prinesle podobnost pri načrtovanju pakirnih linij različnih proizvajalcev. Stroji, zgrajeni po tem standardu, naj bi bili lažji za servisiranje in integracijo v obstoječe linije, ki so zgrajene po tem principu. Koncepti pa se uporabljajo tudi na drugih področjih spremljanja in vodenja procesov, na primer v predelavi surovin, sestavljanju izdelkov, obdelovalnih strojev in v robotiki. Smernice za izvedbo določajo model obnašanja entitet skozi poseben hierarhičen avtomat stanj. Zagotavljajo tudi načine za komunikacijo med različnimi stroji ter med stroji in informacijskimi sistemi.

4 Strojna in programska oprema

Pričujoča aplikacija IIoT vključuje več različnih strojnih in programskih komponent. Proizvodni proces je simuliran znotraj programskega okolja TwinCAT 3.1, ki teče na prenosnem računalniku, s katerim simuliramo programirljivi logični krmilnik (PLK). Z njim se poveže mikroročunalnik Raspberry Pi 3B, ki služi kot robna naprava za obdelavo prejetih podatkov iz PLK-ja in pošilja

nje relevantnih informacij v oblak. Tam smo za demonstracijo uporabili spletno aplikacijo Azure IoT Central, ki omogoča povezovanje, spremljanje in upravljanje z napravami IoT.

4.1 Raspberry Pi

Raspberry Pi je mikroročunalnik, ki je bil razvit v sodelovanju z organizacijo Raspberry Pi Foundation iz Združenega kraljestva za namen spodbujanja poučevanja osnov računalništva med mladimi po svetu. Zaradi majhnosti, nizke cene in zmogljivosti se je začel uporabljati tudi v številnih aplikacijah izven prvotnih namenov uporabe. Do leta 2020 je bilo izdanih že več različnih verzij [2].

V tej aplikaciji je bila uporabljena različica 3B z 1 GB delovnega pomnilnika (RAM), štirijednim 64-bitnim procesorjem, vgrajenim modulom WiFi in Bluetooth, priključkom Ethernet ter štirideset splošnonamenskimi vhodi in izhodi (GPIO).

4.2 Beckhoff TwinCAT

Program TwinCAT 3.1 je namenjen programiranju krmilnikov proizvajalca Beckhoff. Integriran je v okolje Microsoft Visual Studio in temelji na odprtokodnem programskem orodju Codesys. TwinCAT podpira programiranje v jezikih po standardu IEC 61131 [7] (IL, LD, ST, FBD, SFC+CFC). Možna je tudi izmenjava podatkov z moduli, napisanimi v C/C++ ali zgrajenimi v okolju MATLAB/Simulink. Omogoča tudi simulacijo delovanja PLKja.

4.3 Python

Python je odprtokoden, splošnonamenski in zmogljiv programski jezik. Zlasti je v uporabi na področju dela s podatki. Ima široko podporo skupnosti razvijalcev, ki prispevajo programske rešitve za različna področja, tudi v okviru industrije 4.0. Večina osnovnih funkcij je vključenih v jedrni Python modul, dodatni pa so na voljo v obliki knjižnic, ki so javno dostopne. V našem primeru smo med drugim uporabili knjižnico PyAds, ki omogoča povezavo z napravami TwinCAT preko komunikacijskega protokola ADS.

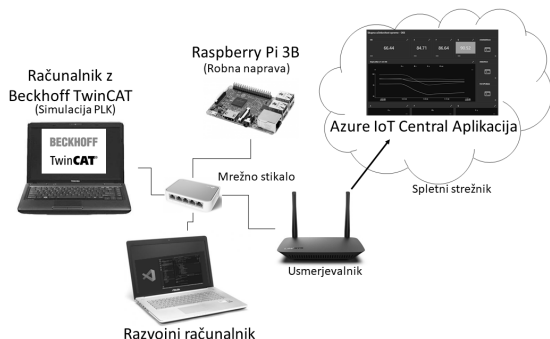
4.4 Azure IoT Central

Azure IoT Central je programska storitev, ki omogoča povezavo, spremljanje in upravljanje naprav IoT v oblaku. Storitve IoT Central je namenjena hitri in enostavnejši vzpostavitvi aplikacij IoT brez skrbi za povezovanje med različnimi potrebnimi komponentami. Vanjo je že vključena komunikacija z napravami, izdelava predlog naprav in hramba ter vizualizacija podatkov.

5 Aplikacija spremljanja učinkovitosti

Aplikacija vključuje tri glavne komponente: PLK, Raspberry Pi in spletno aplikacijo na platformi Microsoft Azure IoT Central. Odločili smo se, da PLK simuliramo znotraj programskega okolja Beckhoff TwinCAT 3.1, torej na osebem računalniku. Računalnik s simuliranim krmilnikom preko kabla CAT5e povežemo na mrežno stikalo, kamor sta priključena tudi Raspberry Pi 3B in dovod iz glavnega usmerjevalnika za dostop do spleta (slika

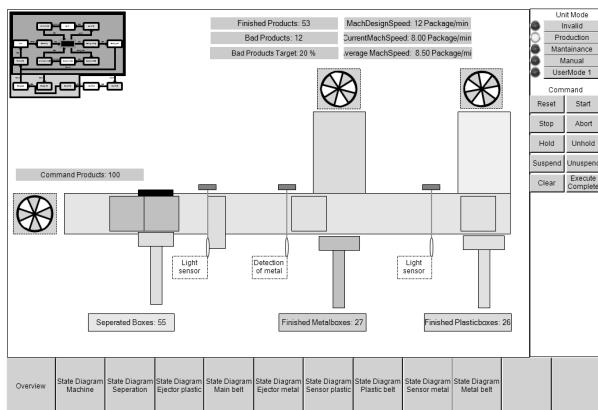
1). Program Python na Raspberry Piju se poveže s PLK-jem preko komunikacijskega vmesnika ADS. Nato smo dodali programsko kodo, ki poskrbi za pošiljanje podatkov v oblak in sprejemanje ukazov, ki jih operater izvrši v spletni aplikaciji Azure IoT Central. Več podrobnosti o izvedbi je mogoče najti v diplomskem delu [8].



Slika 1: Vizualni prikaz testnega sistema.

5.1 PLK

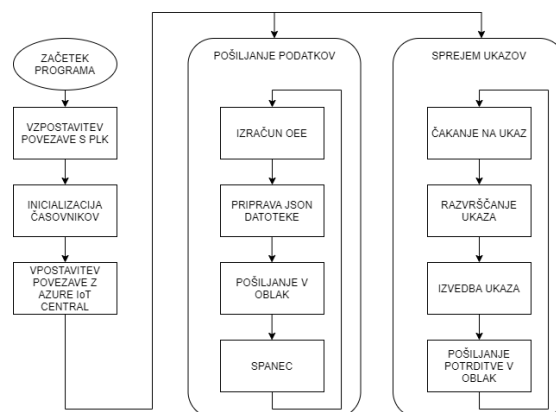
Za simulacijo proizvodnje linije smo uporabili demo primer uporabe knjižnice PackML in ga dopolnili, da ustreza naši aplikaciji. Program predstavlja simulacijo preproste proizvodne linije. Vizualizacija je izdelana interaktivno, tako da v realnem času vidimo premike orodij in izdelkov (slika 2). V zgornjem levem kotu se nahaja diagram prehajanja stanj po modelu, ki ga določa PackML, kjer je z rdečim okvirjem označeno trenutno stanje linije. Na vrhu so prikazani osnovni parametri (št. vseh kosov, št. slabih kosov, trenutna in povprečna hitrost linije), imamo pa tudi možnost nastavitve ciljne hitrosti stroja in odstotka slabih kosov. Na desni strani se nahajajo gumbi za upravljanje s procesom. V spodnji vrstici lahko izbiramo med prikazi diagrama prehajanja stanj za celotno linijo ali samo za posamezna orodja. Demo primer ni imel možnosti izdelave neustreznih izdelkov, zato smo v PLK program dodali funkcijo, ki naključno označuje izdelke kot neustrezne. Njihov delež lahko poljubno nastavimo na osnovnem prikazu delovanja linije.



Slika 2: Vizualizacija simulirane proizvodne linije.

5.2 Robna naprava

Vse spremenljivke, potrebne za izračun kazalnikov OEE, se preko protokola ADS prenašajo v program na Raspberry Piju, napisan v jeziku Python. Tam se izvaja izračun kazalnikov in pošiljanje rezultatov v spletno aplikacijo za nadzor nad OEE v realnem času. Kazalnike bi se lahko izračunavalo tudi na samem PLK-ju, vendar je uporaba vmesne naprave, kot je Raspberry Pi, smiselna v primeru nadzora nad več napravami, saj s tem rezultate različnih strojev ustrezno kombiniramo in v oblak pošiljamo samo podatke, ki so potrebni za pregled sistema iz višjega nivoja. Z uporabo vmesne oz. robne naprave je merjenje OEE enostavneje vpeljati v obstoječe proizvodne procese brez posegov v programe na PLK-jih. Pri izračunu kazalnikov OEE smo si pomagali s spremljanjem časa, ko je bil stroj v posameznem stanju. V Pythonu smo napisali objekt časovnika in ustvarili instanco tega objekta za vsako stanje PackML. Merjenje časa v posameznem stanju se začne oz. konča v primeru spremembe spremenljivke `PackTags.Status.StateCurrent`. Vsi izračuni se hranijo v delovnem pomnilniku Raspberry Pi, dokler program teče. Tak pristop je primeren samo za testiranje, v proizvodnji pa bi bilo potrebno podatke shranjevati v podatkovno bazo.



Slika 3: Poenostavljen diagram poteka programa.

Ko poženemo program, se najprej vzpostavi povezava z napravo TwinCAT (slika 3). Če povezava ni uspešno vzpostavljena, program izpiše napako. Nato se ustvarijo instance časovnikov, ki omogočajo beleženje časa v posameznem stanju. Program takrat že začne beležiti stanje proizvodne linije in pridobivati potrebne podatke za izračun kazalnikov. V naslednjem koraku program vzpostavi povezavo z aplikacijo Azure IoT Central. Zatem vstopimo v asinhrono glavno funkcijo, ki je v grobem sestavljena iz dveh delov. Prvi je namenjen obdelavi in pošiljanju podatkov v oblak, drugi pa sprejemu in izvrševanju ukazov iz oblaka. Oba dela se istočasno izvajata v neskončni zanki.

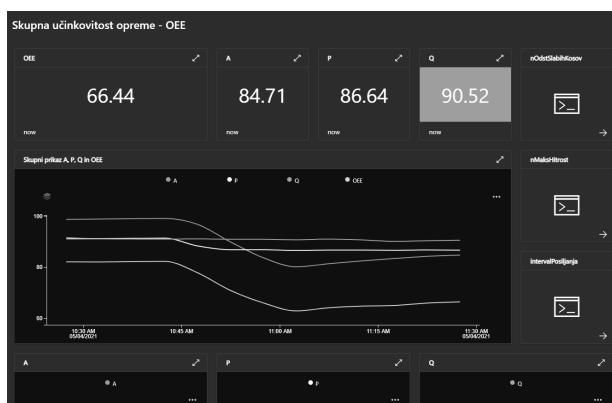
Funkcija za obdelavo in pošiljanje podatkov najprej kliče funkcijo za izračun OEE, ki pridobi podatke, izračuna kazalnike in pripravi datoteko JSON za prenos v oblak. Nato gre funkcija v asinhrono spanec, ki ne blokira izvajanja ostalega dela programa. Po poteku določenega časa

se proces ponovi.

Sprejem ukazov je sestavljen iz asinhronih funkcij, ki ves čas čaka na ukaz iz aplikacije. Ob sprejemu novega ukaza ga razvrsti glede na ime in podatke v njem ter izvrši določeno akcijo. Zatem je potrebno poslati nazaj potrditev o sprejemu in obdelavi ukaza.

5.3 Spletna aplikacija

V Microsoft Azure IoT Central smo ustvarili novo aplikacijo, kamor smo dodali našo robno napravo. Izračun podatkov se izvede glede na nastavljeno periodo (npr. vsako minuto), nato pa sledi prenos v spletno aplikacijo. Rezultati so prikazani v številski obliki (zadnja znana vrednost) in v obliki grafov, kjer lahko spremljamo zgodovino gibanja kazalnikov (slika 4). Za lažji nadzor smo v aplikaciji nastavili tudi barvno označevanje vrednosti, ki so manjše oz. večje od določene meje. Kazalnike smo sprva želeli prikazati v obliki kazalčnih instrumentov, vendar Azure IoT Central tega prikaza ne omogoča. Tudi sicer je nabor prikazov dokaj omejen, sploh v primerjavi z nekaterimi odprtokodnimi orodji za vizualizacijo podatkov (npr. spletno programsko orodje Grafana). Implementi-

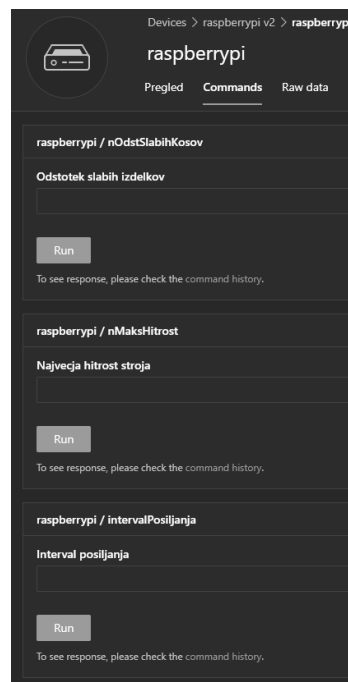


Slika 4: Nadzorna plošča v spletni aplikaciji.

rali smo tudi komunikacijo od oblaka proti robni napravi, s katero lahko v spletni aplikaciji nastavljamo parametre na Raspberry Piju ali vrednosti spremenljivk v programu PLK. Dodali smo možnost nastavljanja periode pošiljanja podatkov v aplikacijo in nastavljanje vrednosti spremenljivk za maksimalno hitrost stroja ter odstotek slabih izdelkov v simulaciji (slika 5). Vsakemu ukazu mora slediti odgovor, da operater ve, ali je naprava ukaz sprejela in ga ustrezno obdelala. Če odgovora ni, lahko operater hitro ukrepa in pošlje nekoga v pregled stanja na terenu.

6 Zaključek

Ugotovili smo, da je obdelava podatkov na robu omrežja smiselna, sploh v primeru večjega števila povezanih naprav in velike količine podatkov. Uspešno smo izdelali aplikacijo, ki bi bila z nekaj dopolnitvami primerna tudi za vključitev v realne proizvodne procese. Tak primer izboljšave bi bila dodatna podatkovna baza na robu, ki bi premoščala težave ob morebitnih izpadih komunikacije med robno napravo in PLKjem ali pri nenadnem izklopu



Slika 5: Prikaz ukazov v spletni aplikaciji.

robne naprave. Za lažjo konfiguracijo nastavitvev, bi bilo smiselno dodati uporabniški vmesnik, na katerem bi nastavljali parametre za povezavo med napravo in oblakom. Kar se tiče spletnih orodij Azure IoT, smo ugotovili, da so zaradi odličnih možnosti integracije primerna predvsem za uporabnike, ki že uporabljajo ostale Microsoftove storitve (npr. Power BI). Zaradi pomanjkanja fleksibilnosti pri izdelavi vizualizacije bi bilo zanimivo poskusiti z izdelavo podobne aplikacije, ki pa bi temeljila na odprtokodnih rešitvah (npr. kombinacija podatkovne baze InfluxDB in spletne aplikacije Grafana).

Literatura

- [1] F. Al-Turjman, *Edge Computing: From Hype to Reality*. ZDA: Springer International Publishing, 2019.
- [2] *Raspberry Pi*. Dosegljivo: https://sl.wikipedia.org/wiki/Raspberry_Pi. [Dostopano: 12. 7. 2021].
- [3] Ki-Young Jeong, D. T. Phillips, "Operational efficiency and effectiveness measurement" v *International Journal of Operations & Production Management*, vol. 21. 2001, str. 1404-1416.
- [4] *PackML*. Dosegljivo: <https://en.wikipedia.org/wiki/PackML>. [Dostopano: 12.7.2021].
- [5] "Machine and Unit States: An Implementation Example of ANSI/ISA-88.00.01", ANSI/ISA, TR88.00.02, 2015.
- [6] "Batch Control Part 1: Models and Terminology", ANSI/ISA-88.00.01, 2010.
- [7] "Programmable controllers - Part 3: Programming languages", IEC 61131-3, 3., 2013.
- [8] J. Špeh, "Spremljanje učinkovitosti proizvodnih procesov z obdelavo na robu omrežja", diplomsko delo, Ljubljana, Univerza v Ljubljani, Fakulteta za elektrotehniko, 2021.