

Distributed Computing in ZDMP

By Tim Dellas. Ascora GmbH

Some questions for you

- Do you want to store machine generated data in the Cloud, or on-premises?
- When every device has a computer in it, where should the computation of sensor data occur? In the Cloud? Or close to the device that produces the data? Or maybe close to the device that needs the data?
- Is Edge management a problem that only specialists can solve, or could there be user-friendly web tools that enable unspecialized workers to configure Edge resources?
- When every device has a computer with a different purpose and operating system, how can you make sure that platform services can run on all of those?

Motivation

In traditional manufacturing all calculations were done in the data centre of a company, or directly at the machines, where data was obtained. Long before Industry 4.0 interconnection trend emerged. As more and more computers crept into manufacturing, customer relationship systems, production planners, workforce management systems etc the problem of interconnecting these in a useful and secure fashion has arisen. Finally, Industry 4.0 and its many manifestations became a difference maker in the effectiveness, productivity and defects minimization in manufacturing.

With manufacturing platforms appearing, all equipment on the shopfloor has started to rely more and more on common protocols. As a consequence, whole new world of applications, optimization and interconnectedness has emerged. To make these boundless possibilities graspable for industry users, platforms begin to employ an app-concept, and treat the factory as a smart computer with lots of moving parts, using the platform itself as an operating system. Putting functionality into zApps is exactly the concept implemented by ZDMP.

This comes with its own share of problems though. Relaying computation to a Cloud implies that data could in theory leak to other parties in case of a security breach, and data can give away a company's intellectual property. Many manufacturers are not IT specialists and cannot evaluate the threat that this poses to their enterprise. ZDMP is a platform answering this need for control, security, user management, and encryption in an appropriate way.

What does Distributed Computing mean in the context of ZDMP?

The same lack of IT knowledge means that the employees of a manufacturing company, whose expertise lies in the manufacturing subject matter, cannot implement many of the technical necessities of the new conceptual model of Industry 4.0. ZDMP also helps out in these computing questions, where either it might be acceptable to compute in the Cloud, but where also still the quickest path from machine to computing node might be important and where the computing node should be close to the machine (at Edge level). ZDMP will empower users with a simple web-based tool to control where which service is running – without the need for specialized IT staff.

Distributed Computing can have many meanings – in the area of Industry 4.0 it might mean the computational distribution of complex calculations to parallelize them on different machines, it might also mean discussion of the factory hall wiring/interconnection or industrial Ethernet protocols etc.

In the context of ZDMP, the Distributed Computing task deals with two things: the first is the software management on computing nodes from spatial perspective, ie. Cloud, Fog, Edge and Mist; and the other related to the running of deployable services on hardware that might be limited in terms of memory and computational resources.

Platform Services running in the Cloud?

Talking about Industry 4.0 always means talking about the Cloud, and the level where the platform runs. [The predecessors of ZDMP found that industry is not very fond about transferring](#) data, which might contain business secrets, away from the factory's side into a 3rd party server. Manufacturing companies sometimes expect that a platform can completely run on the company's premises, without a chance of data leakage.

Layered physical node separation by context

Whilst the **Cloud** layer is where the default components from ZDMP are running, the layer of the company's premises is called a **Fog** layer. The management of the physical locations/spatial aspects of computing nodes and machines is performed within **Edge** layer, as sometimes data need to be quickly computed close to the machine that produces data, or a service might need to be in close range to moving devices such as shopfloor robots. Distributed Computing task in ZDMP is in charge of creating the rules on which level and, if not in the Cloud, on which devices a service should run.

Virtualization Strategy within ZDMP

And with all these different machines, the question arises which service has to be run on which kind of hardware and on which kind of operating system. ZDMP also wants to help the users in this regard and uses an abstraction layer to be able to run any service, no matter which programming language and frameworks were used, on any kind of hardware and software infrastructure. [Kubernetes](#) is the chosen technology for managing this. Kubernetes is the most advanced and configurable industry solution to virtualize services on computers to separate the physical layer from the operating system and the file system. In this way, all the components and ZDMP Applications (zApps) (which have to use Docker as a container technology) running on the platform can use the central Kubernetes instance and find themselves deployed in the platform, no matter which programming languages or frameworks were used to create them. During the last year the ZDMP project became aware that lightweight devices might be too underpowered to run a local Kubernetes client, so the Distributed Computing task was selected to create another platform layer, which is supposed to only include edge devices using a lightweight implementation of Kubernetes named K3S. The graphical user interface, described in the section above, will be used to configure which services to run and in which locations selecting between the available nodes, while a default installation of Kubernetes will do the assignment of the services towards nodes. A descriptive path will be the location identifier, like "Ascora/HQ/Building1/Level1/Wing3/Room1". When the deployment scheme works, the main Kubernetes cluster from the Platform and the Distributed Computing tasks will be combined in a holistic system.

What will ZDMP achieve

ZDMP will manage edge nodes with an application-level user interface and use Kubernetes and K3S for the edge devices. An application level logical data structure is used to define spatial characteristics and model these in a tree-structure for Kubernetes, using tags of the single nodes to identify their location. For 3rd party developers, a generic template for services contains the basic configuration files and HELM charts, and just needs to be fully configured to guarantee ZDMP compatibility. A tutorial documentation is created for the setup of K3S instances on smaller computing nodes, as these need to be prepared for enabling ZDMP components or zAssets to run.

ZDMP Links

| | |
|------------------------------------|---|
| • Architecture Component(s) | Distributed Computing |
| • Work Package | WP5 – ZDMP Core Services |
| • Tasks | T5.5 – Distributed & Autonomous Computing |