

Systemarchitektur eines Sensor/Aktor Knotens für dezentralisierbare Aufgabenverteilung

Aufgabenstellung

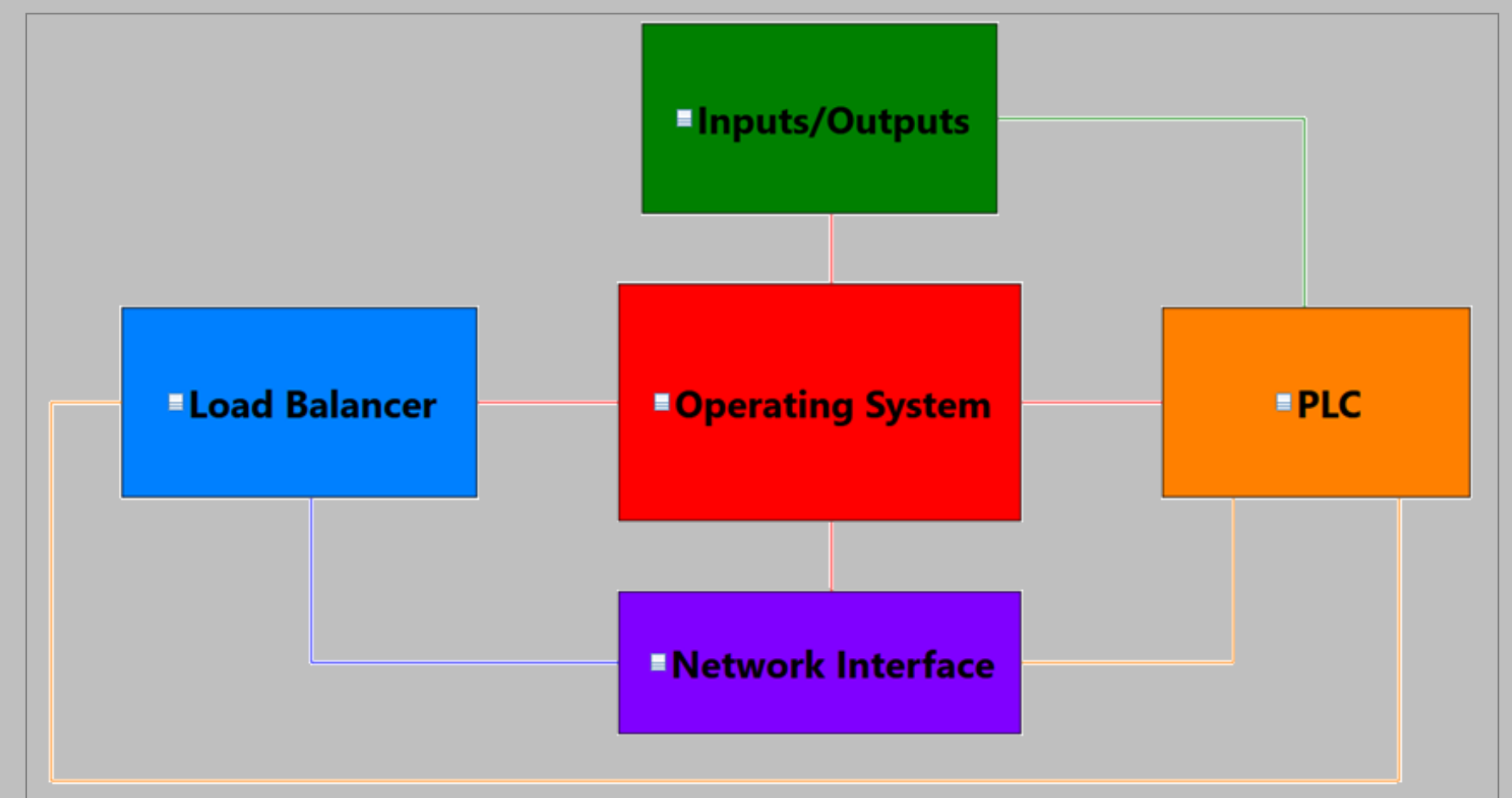
Vordefinierte Load Balancing Tasks in einem Netzwerk aus Sensor/Aktor Knoten sollen dynamisch auf verschiedene CPUs aufgeteilt werden. Die Auslagerung erfolgt nach taskspezifischen Kriterien sowie der Auslastung des Knotens selbst. Zu diesem Zweck wird die Softwarearchitektur des Knotens entwickelt und modelliert.

Eine besondere Schwierigkeit ist dabei die Vereinigung von zeitkritischer Aufgabenverteilung, dynamischer Netzwerkkommunikation und Echtzeitbearbeitung der Ein- und Ausgänge.

Damit die Portierung des Softwaregerüsts auf andere Plattformen zu einem späteren Zeitpunkt gewährleistet werden kann, soll die Architektur möglichst abstrakt beschrieben werden.

Modellierung

Die Modellierung erfolgt in UML (Unified Modeling Language). Da die Netzwerkarchitektur gleichberechtigt und verteilt sein soll, werden auf jedem Knoten ein OPC UA Server und Client installiert. Das vereinfachte Class Diagram des Modells stellt alle benötigten Komponenten eines Knotens dar.

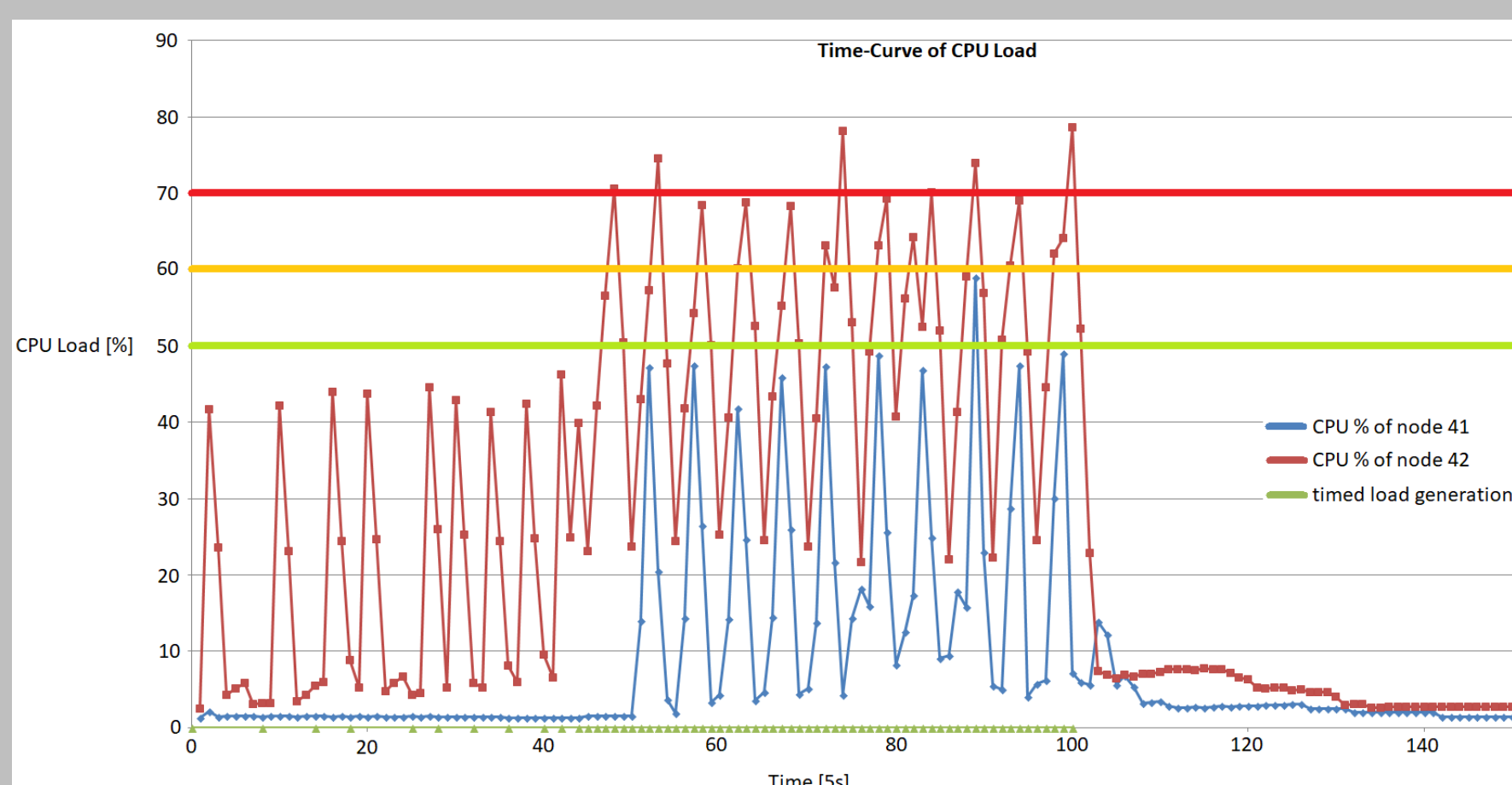


Technologien und Umsetzung

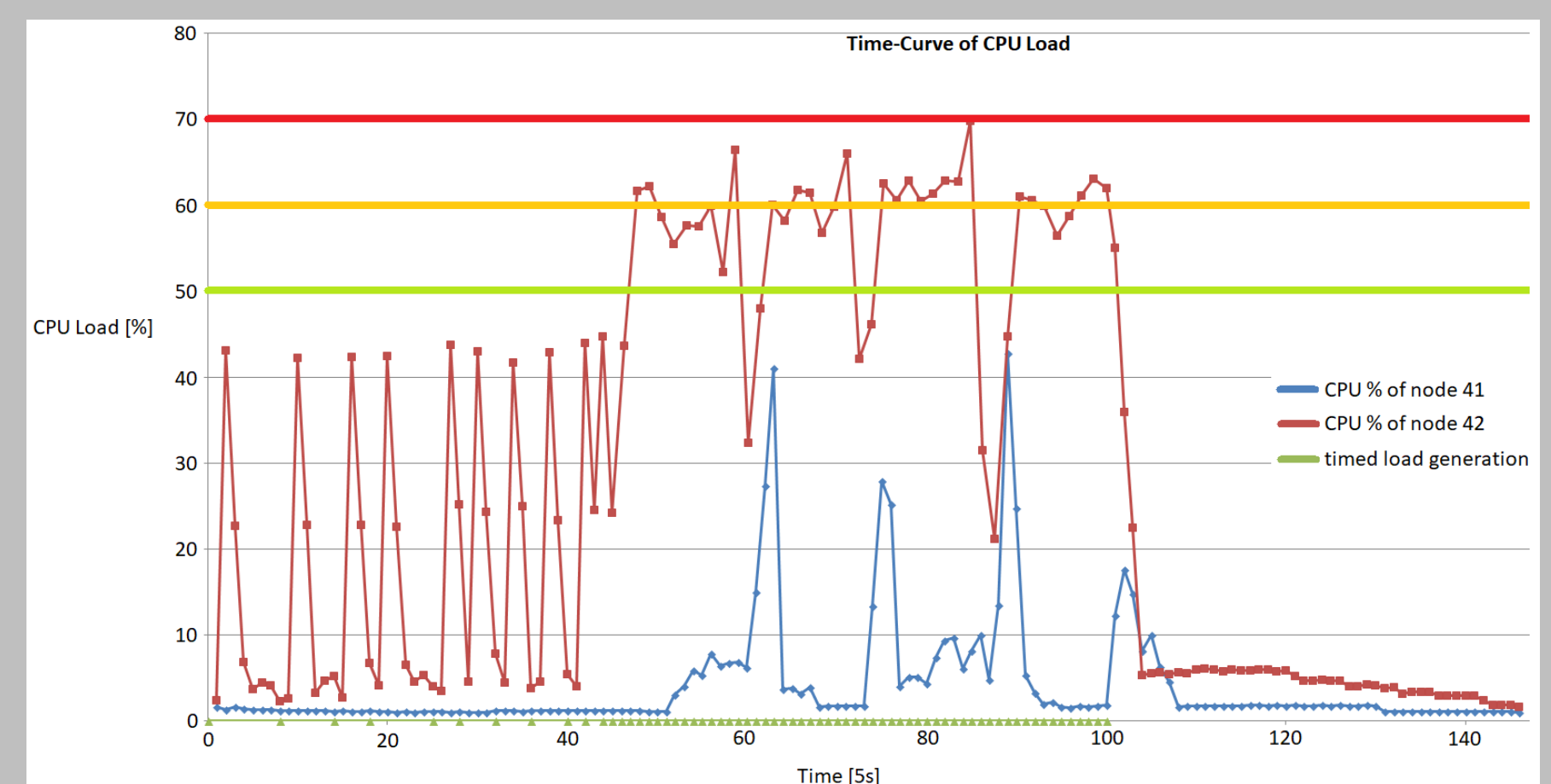
Die Umsetzung ist zur Gänze auf offenen Standards und lizenzfreien Komponenten aufgebaut. Zur Kommunikation zwischen den Knoten wird der Open Source Stack open62541 verwendet. Als Betriebssystem auf dem BeagleBone Black wird die freie Distribution Linux Debian installiert. Zur Umsetzung werden die modellierten Softwarekomponenten auf zwei Entwicklungsboards in C und C++ programmiert. Dabei generiert ein Knoten die Last, der zweite Knoten nimmt die Überlast des ersten Knotens an.



Ergebnis



Mittelwertbildung über 5 Aufrufe



Mittelwertbildung über 10 Aufrufe

Während der Laufzeit des Systems wird die CPU Last auf 2 Knoten aufgezeichnet. Die rote Lastlinie ist die der belasteten CPU, die blaue Kennlinie ist die Last der CPU, welche die Tasks annimmt. Die Lastmessung wird über 5 und 10 Aufrufe gemittelt. Durch die längere Messung wird die Zeitkonstante des Systems erhöht, wodurch es zu geringeren Spitzen in der Auslastungsmessung kommt, und die CPU Last ist leichter zu kontrollieren.

