Optimalizační úlohy v hierarchických decentralizovaných řídicích systémech (3)

Zdeněk Bradáč, Václav Jirsík

2.5 Formální model požadavků programového vybavení decentralizovaného řídicího systému

Vzhledem k zavedenému formálnímu popisu hardwarové struktury hierarchického decentralizovaného řídicího systému za pomocí orientovaného ohodnoceného grafu je nezbytné využít stejný formální popis i pro charakterizování vlastností programového vybavení decentralizovaného řídicího systému.

Nezbytným předpokladem k formálnímu popisu programového vybavení je striktní dodržování zásad modulového členění programového vybavení. Pro formální popis programového vybavení je nezbytné definovat strukturu programového vybavení tak, aby bylo možno separovat programové moduly a definovat k nim korespondující požadavky na datovou výměnu mezi jednotlivými bloky. V první fázi formálního popisu programového vybavení je nezbytná znalost následujících parametrů:
- počet programových a organizačních bloků,
- požadavky každého programového bloku na HW zdroje,
- požadavky každého programového bloku na datovou výměnu s dalšími bloky.

V této fázi není nezbytná znalost vnitřní struktury programových a organizačních bloků, ale je nezbytná znalost jejich návazností na své okolí. K programovému vybavení decentralizovaného systému potom přistupujeme z globálního pohledu, přičemž nemusíme programovat jednotlivé HW komponenty decentralizovaného systému, ale můžeme vytvářet globální program skládající se z komunikujících funkčních a organizačních bloků, případně funkcí. Takovouto strukturu celkem vhodně definuje norma IEC 1131, primárně její třetí část.

Následně pro grafický model programového vybavení decentralizovaného systému využijeme opět grafový formalismus, přičemž jednotlivé vrcholy budou představovat programové moduly a orientované hrany budou představovat požadavky na datovou výměnu mezi jednotlivými programovými moduly.

Ke každému vrcholu je přiřazena množina požadavků odpovídajícího programového modulu na využití HW zdrojů. Množina se může skládat z prostých numerických kvantifikátorů ve formě vektorů či matic. V případě složitějších požadavků je možno akceptovat jejich vyjádření výčtem či jinou formou.

Orientovaná hrana vyjadřuje směr požadovaného toku dat a přiřazená množina požadavků umožňuje kvantifikovat datové požadavky. V mnoha případech množina přiřazená hranám se degraduje na prosté numerické kvantifikátory. Vzhledem k tomu, že datová výměna může být co do datové struktury dosti složitá, což by mohlo vést k definici každého komunikačního vlákna (každé proměnné) jako samostatné hrany v grafu, je vhodné přistoupit k generalizaci datových požadavků.

Generalizace datových požadavků umožní sloužení všech datových požadavků, které mají společný zdroj a cíl v jeden generalizovaný datový požadavek. Toto řešení umožní značně omezit počet datových požadavků, což má podstatný význam v redukci nezávislých proměnných matematického modelu definovaného v následujících kapitolách.

V případě, že množiny přiřazené vrcholům a hranám obsahují pouze numerické kvantifikátory je možno pro řešení optimalizačních úlohy využít numerických metod. V ostatních případech je nutno volit jiné metody řešení, jako například metody umělé inteligence, genetické algoritmy či jiné metody.

3. Matematický model

Matematický model vychází ze zásad lineárního programování (LP) a umožňuje převést grafický model hardwarové struktury hierarchického decentralizovaného řídicího systému a taktéž grafický model programového vybavení do matematické reprezentace. Tato matematická reprezentace se obecně skládá ze soustavy rovnic, omezujících podmínek a kritérií, na základě kterých je možno řešit optimalizační úlohy spojené s konfigurací decentralizovaného řídicího systému. V zásadě výše uvedené grafické modely umožňují řešit dva základní problémy související s konfigurací, a to:
- optimal routing problem – úloha zaměřená na optimální směrování a routování datových komunikačních paketů v rámci komunikační struktury hierarchického decentralizovaného řídicího systému,
- optimal mapping problem – úloha zaměřená na optimální pokrytí HW zdrojů programovými moduly tak, aby bylo splněno definované kritérium. Jedná se vlastně o matematické přiřazení programových modulů jednotlivým HW zdrojům, na kterých budou v době běhu aplikace spouštěny.

Pro matematický model související s řešením úloh optimálního směrování komunikačních paketů se vychází ze zásad klasické úlohy LP, avšak pro řešení problému optimálního mapování programových bloků je nezbytné vycházet z rozšířené úlohy lineárního programování, tzv. MILP (mixed integer linear programming).

V případě mapovacího problému se vychází z řešení úlohy celočíselného lineárního programování, přičemž se úloha ještě komplikuje, jelikož celočíselné proměnné degradují na proměnné binární, tzv. optimální přiřazovací problém.

Z tohoto důvodu se následující kapitoly budou zabývat čistě úlohou optimálního směrování datových paketů v rámci decentralizovaného systému. Úloha optimálního mapování programových bloků bude degradována na prosté explicitní přiřazení bez nutnosti řešení optimalizační úlohy.

3.1 Optimální směrování paketů v rámci decentralizovaných řídicích systémů

Matematický model vhodný pro řešení optimalizační úlohy zaměřené na optimální směrování datových paketů v rámci decentralizovaných řídicích systémů je založen na vytvoření soustavy matematických rovnic, soustavy omezujících podmínek založených na definici soustavy nerovnic a optimalizačního kritéria.

Matematický model je založen na grafickém modelu hardwarové struktury decentralizovaného řídicího systému a na grafickém modelu programového vybavení. Oba grafy (kapitola 2) jsou definovány jako orientované vážené grafy, přičemž veškeré váhy a přiřazené vektory obsahují pouze numerické kvantifikátory.

3.2 Grafová prezentace HW struktury decentralizovaného řídicího systému

Nechť je hardwarová struktura decentralizovaného systému popsána orientovaným váženým grafem (obr. 10). Graf obsahuje pouze dva typy objektů, a to vrcholy a orientované hrany. Vrcholy grafu popisují jednotlivé hardwarové zdroje ve struktuře decentralizovaného řídicího systému. Každý vrchol je charakterizován vektorem, přičemž tento vektor může popisovat schopnosti HW zdroje, např.:
- výpočetní kapacitu,
- numerickú výpočetní kapacitu,
- kapacitu operační paměti,
- kapacitu paměti programu,
- kapacitu permanentních pamětí,

atd.

V tomto případě jsou všechny členy vektoru numerického charakteru a dají se numericky kvantifikovat. Tento vektor je přiřazen každému vrcholu v grafické prezentaci.

Obdobně hrany v grafické prezentaci charakterizují následující přenosové schopnosti průmyslových sériových komunikačních sběrnic:
- přenosová kapacita,
- cena za přenesení datové jednotky.

Situace znázorněná na obr. 10 popisuje systém se čtyřmi hardwarovými zdroji (počítači), které jsou vzájemně propojeny šesti komunikačními sběrnicemi.

3.3 Formální model požadavků programového vybavení

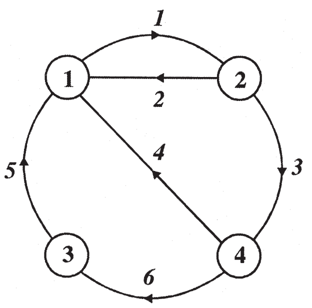
Podmínkou pro strukturu aplikačního programového vybavení je její striktní modularita. Vlastní aplikace je definována jako soubor programových modulů (organizační moduly, výkonné moduly, funkce, atd.), které vzájemně mohou komunikovat.

Každý programový modul může komunikovat s kterýmkoliv dalším programovým modulem, avšak musí být exaktně zadefinován datový tok, který je orientován (obr. 11). Jednotlivé vrcholy formálního model požadavků programového vybavení popisují požadavky jednotlivých programových modulů. Je to např.:
- požadavek na výpočetní výkon,
- požadavek na operační paměť,
- požadavek na datovou paměť,
- požadavek na výpočetní výkon v plovoucí řádové čárce,

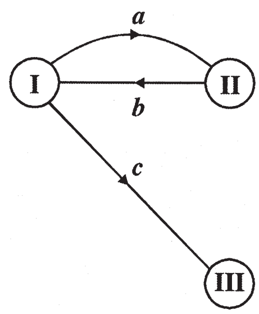
atd.

Každá hrana definuje požadavek na komunikaci mezi dvěma programovými moduly (zdrojovým a cílovým modulem). Každá hrana definuje:
- požadavky na směr komunikace při přenosu dat mezi dvěmi programovými moduly,
- váhu, která numericky kvantifikuje datový tok přes danou hranu.

Situace na obr. 11 znázorňuje programové vybavení sestávající ze tří programových bloků, které vzájemně komunikují přes tři datové toky. Datové toky jsou popsány numerickým vektorem $\{a, b, c\}$.



Obr.10 Grafová prezentace HW struktury DCS



Obr.11 Formální model požadavků programového vybavení

3.4 Matematický model HW struktury decentralizovaného řídicího systému

Grafová reprezentace HW struktury DCS může být přepsána do souboru lineárních rovnic. Všechny rovnice jsou konstruovány na základě zásad lineárního programování. V první fázi je nezbytné definovat stavové proměnné grafu. Proto datový tok přes každou hranu i nechť je popsán stavovou proměnnou x_i. Následně pro každý vrchol j grafu může být sestavena lineární rovnice takto:

$$\forall j : \sum_i x_i = 0 \qquad (2)$$

pro $j = \{1, 2, 3, 4\}$ a $i = \{1, 2, 3, 4, 5, 6\}$

Takže pro každý vrchol (obr. 10) platí lineární rovnice:

vrchol č. 1: $-x_1 + x_2 + x_4 + x_5 = 0$

vrchol č. 2: $x_1 - x_2 - x_3 = 0$

vrchol č. 3: $-x_5 + x_6 = 0$

vrchol č. 4: $x_3 - x_4 - x_6 = 0$

Soustavu lineárních rovnic zapíšeme maticově:

$$\begin{pmatrix} -1 & 1 & 0 & 1 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 & 0 & -1 \end{pmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (3)$$

Z maticového zápisu extrahujeme incidenční matici grafu:

$$A = \begin{pmatrix} -1 & 1 & 0 & 1 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 & 0 & -1 \end{pmatrix} \qquad (4)$$

Jelikož je jeden řádek incidenční matice lineární kombinací ostatních, potom libovolný řádek incidenční matice může být vyjmut ze soustavy lineárních rovnic. V našem případě byl zvolen řádek č. 3 k vyjmutí:

$$A = \begin{pmatrix} -1 & 1 & 0 & 1 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{-1} & \cancel{1} \\ 0 & 0 & 1 & -1 & 0 & -1 \end{pmatrix} \qquad (5)$$

Nyní získáme redukovaný systém rovnic vyjádřený redukovanou incidenční maticí. Redukovaný systém lineárních rovnic je následující:

$$\begin{pmatrix} -1 & 1 & 0 & 1 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \end{pmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (6)$$

Pokračovanie v budúcom čísle.

Ing. Zdeněk Bradáč
Ing. Václav Jirsík, CSc.
VUT FEKT Brno
Božetěchova 2, 612 66 Brno, ČR
e-mail: bradac@feec.vutbr.cz
jirsik@feec.vutbr.cz