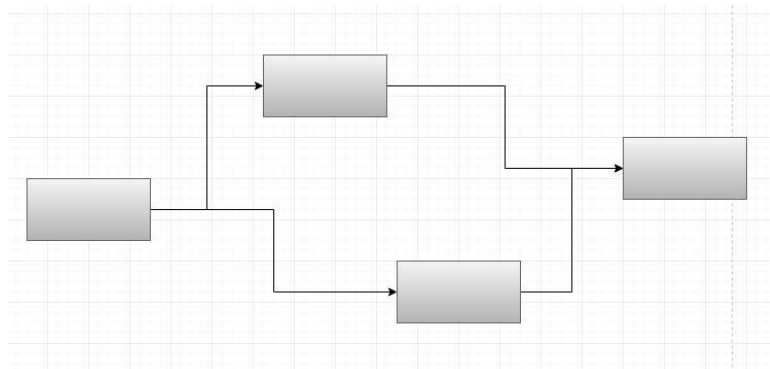
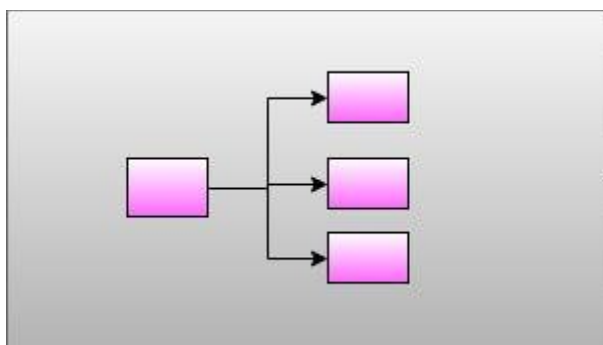


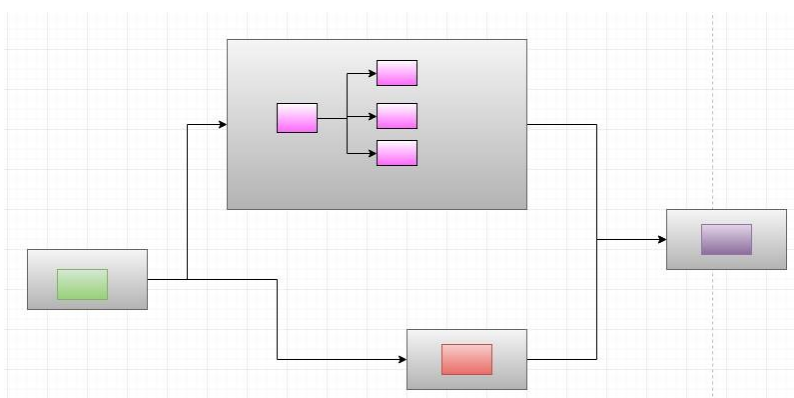
Szóval alapvetően így néz ki a Pipeline, azaz néhány Node dependency gráfba kötve:



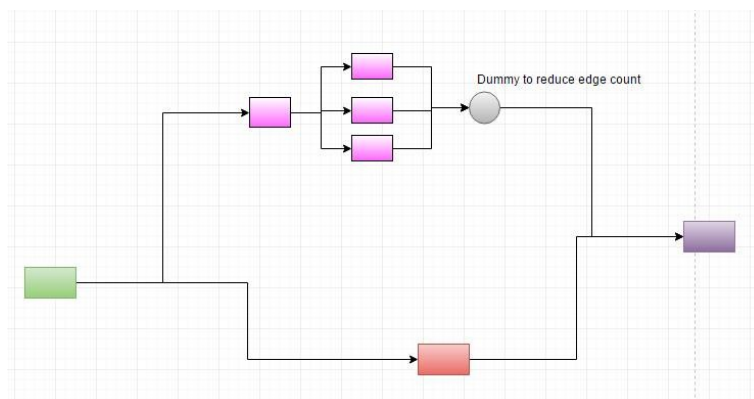
Minden Node visszaad egy `gráf<std::function<>>` cuccot, aminek a neve Task. Itt egy 4 functionből álló példa van, amiből egy inicializálás, a másik 3 pedig párhuzamos meló.



Ha a node-oknak jelöljük a taszkjait, akkor valahogy így fog kinézni a teljes pipeline:



Ezt csak úgy lehet végrehajtani, ha kibontjuk egy nagy, globális `std::function` gráfra az egészet:



A példán jól látható, hogy a párhuzamos és multi-gpu taszkokat hogy lehet megoldani, illetve hogy a Node-ok többsége marad majd az egyszerű 1:1 node:function módszernél.

Ami a szinkronizálást illeti, felmerül az a probléma, hogy egy Node async computeot vagy async copy-t szeretne indítani. Az async compute tipikusan inkább másik Node lesz, de a copy az egy elég tipikus dolog lehet. Az a terv, hogy a Task std::function grájfjával le tudja írni majd a Node, hogy milyen módon kell majd szinkronizálni az async copy-t a többi dologgal, és akkor a Node-nak nem kell a fencekhez nyúlnia. Ez lényegesen egyszerűsíti a dolgokat a Node-ok tervezésénél, mert a dependency gráf sokkal intuitívabb, mint a fence értékek kezelése.