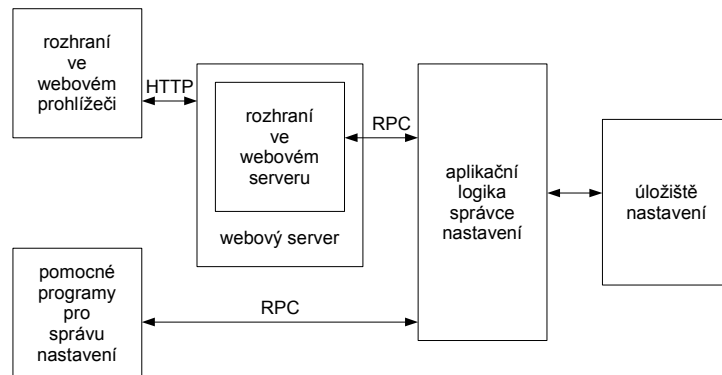


Cílem vaší práce je navrhnout middleware pro aplikaci s three tier architekturou na přiloženém obrázku. Touto aplikací je imaginární internetový správce uživatelského nastavení programů, jehož funkce jsou následující:

- Schopnost evidovat pro každého registrovaného uživatele sadu nastavení programů v podobě archivu souborů s nastavením. Formát souborů s nastavením není správci obecně známý, správce umí pouze evidovat archivy souborů s nastavením, u každého archivu si pamatuje:
 - název programu, ke kterému archiv patří,
 - počítač, kde byl archiv pořízen,
 - datum, kdy byl archiv pořízen.
- Schopnost uložit na požádání nové nastavení programu.
- Schopnost vydat na požádání nastavení programu podle jména programu a dalších podmínek:
 - nejnovější nastavení programu,
 - o jedno starší než nejnovější nastavení programu,
 - nejnovější nastavení programu z konkrétního počítače.



Správce uživatelského nastavení programů bude nabízet jednak webové rozhraní, na kterém uživatel uvidí přehled svých nastavení a bude si moci stáhnout vybraný archiv souborů s nastavením, a jednak programové rozhraní, pomocí kterého budou moci funkce správce používat programy

na počítači uživatele (podle situace přímo uživatelské programy, jejichž nastavení se spravuje, nebo pomocné programy, které spravují nastavení a uživatelské programy spouští).

Podmínky

Návrh middleware musí splňovat technické podmínky kladené jednotlivými body zadání. Pokud se budete domnívat, že některé technické podmínky zadání brání rozumnému řešení, můžete si podmínky zadání upravit, takovou úpravu je však nutné dobře zdůvodnit.

- Návrh middleware může kopírovat nebo používat stávající technologie, ale v případě, že je vyžadován popis funkce, musíte funkci těchto technologií popsat.
- Návrh middleware můžete před termínem zkoušky diskutovat, ale vaše řešení musí být originální.
- K vypracování řešení na termínu zkoušky nesmíte používat dříve připravené textové materiály.

Hodnocení

Návrh bude hodnocen podle toho, jak dobře vychází vstříc praktickým požadavkům aplikace, mezi které patří elegance a jednoduchost návrhu a udržitelnost a škálovatelnost implementace. Každá část návrhu bude hodnocena body takto:

- 0 bodů za v principu použitelné řešení,
- 1 bod za obvyklé použitelné řešení,
- 2 body za kvalitní použitelné řešení,
- mínus 1 bod za nefunkční řešení,
- mínus 2 body za beznadějně nefunkční řešení.

U každé části je navíc možnost udělení kladného bonus bodu. Za úspěšné bude považováno řešení s nezáporným počtem bodů. Za výborné bude považováno řešení s průměrným počtem 2 body za každou část.

1. Architektura

K náčrtku architektury správce nastavení doplňte seznamy existujících middleware technologií, které by mohly být architekturou vhodně použity. Seznam doplňte vysvětlením, jakou funkci by konkrétní middleware technologie poskytovaly.

Hodnocena bude zejména správná identifikace všech míst, kde může middleware pomoci, přiměřeně také zda zvolený middleware nabízí potřebné funkce.

2. Rozhraní

Navrhněte rozhraní aplikační logiky správce nastavení, které bude voláno pomocí RPC. Vedle zřejmých funkcí předpokládejte také, že pomocné programy pro správu nastavení se budou moci u aplikační logiky správce nastavení zaregistrovat tak, aby jim správce po nějakou dobu posílal upozornění na změny nastavení.

Hodnocena bude zejména elegantnost rozhraní při volání pomocí RPC a poskytování všech funkcí správce nastavení.

3. Komunikace

Vyberte z vašeho rozhraní tu sadu funkcí, která dovoluje přidat do evidence nový program a uložit a přečíst jeho nastavení. Napište pro tyto funkce sadu stubů na straně klienta a serveru, které zprostředkují volání pomocí RPC. Předpokládejte existenci rozumných komunikačních funkcí jako je spolehlivé odeslání a příjem zprávy, vytváření a rušení zprávy, přístup k obsahu zprávy a podobně.

Hodnocena bude zejména úplnost předávané informace. Samotný kód stubů může být psán heslovitě, očekává se text, podle kterého rozumný programátor vytvoří funkční program, nikoliv text, který je přímo funkční program.

4. Paralelismus

Následující otázky se týkají modelu paralelismu, kterým se rozumí sada pravidel definujících kdy a kde jsou vytvářena, používána a rušena vlákna aplikace, případně kdy a kde jsou zamykány a odemykány důležité datové struktury aplikace. Zde se věnujte pouze vláknům, nikoliv zámkům.

Hodnoceno bude pochopení vztahů mezi technickými aspekty návrhu, jako jsou existence vláken a zámků či rozsah zamykaných dat, a uživatelskými aspekty návrhu, jako jsou schopnost rychlé odezvy a schopnost práce pod zátěží.

4.1. Navrhněte rozumný model paralelismu v aplikační logice správce nastavení tak, aby podporoval připojení více klientů.

4.2. Navrhněte rozumný model paralelismu v pomocných programech pro správu nastavení tak, aby bylo podporováno zasílání upozornění na změnu nastavení.
