Frequent Pattern Mining yang mengadopsi pendekatan apriori memiliki peforma yang bagus dengan mengurangi secara signifikan himpunan kandidat yang ada. Namun, teknik ini memiliki beberapa masalah jika diterapkan pada pola data yang panjang, profilic atau pola yang memiliki treshold support yang rendah. Masalah tersebut sebagai berikut:

1. Ada jumlah himpunan kandidat yang besar, pendekatan ini akan memakan sangat banyak sumber daya.
2. Jumlah waktu yang diperlukan pada pola data yang panjang akan sangat lama. Hal ini terjadi karena terjadinya bottleneck pada tahap candidate set generation dan tes. Hal ini bisa diatasi dengan menggunakan struktur data yang berguna untuk menghindari penyusunan himpunan kandidat yang besar

Berdasarkan paper, berikut merupakan langkah/tahapan dalam melakukan mining secara efisien:

1. Buat sebuah data struktur yang padat yang akan disebut dengan frequent pattern tree (fptree). fptree ini merupakan prefix-tree yang menyimpan data krusial dan quantitatif dari pola data. Data yang sering muncul akan memiliki peluang node untuk terhubung dengan node lain dibanding dengan data yang jarang keluar.
2. Mining pattern pada tree ini dilakukan secara rekursif di dalam tree ini sendiri dengan dimulai dari panjang pola ke -1 dan bergerak sesuai sub pola dari data.
3. Pola pencarian dalam tree menggunakan teknik partisi divide and conquer untuk mengeliminasi subtree yang tidak diperlukan dalam pencarian.

Dari struktur data baru ini kita mendapati beberapa keunggulan di antaranya adalah:

1. Struktur data ini menyusun sebuah tree yang padat informasi dan lebih kecil dari database aslinya.
2. Struktur data ini menghindari pembentukan himpunan kandidat dengan jumlah besar sehinggga bisa memperkecil pilihan yang tidak perlu untuk diproses pada proses mining.
3. Penerapan pencarian partisi divide and conquer secara dramatis mengurangi subsekuen conditional pattern base dan conditional fptree.

Berdasarkan performa studi pada paper, metode tersebut dapat melakukan mining dengan pattern yang pendek maupun panjang secara efisien di basis data yang besar. Hal tersebut menunjukan performa yang signifikan dibandingkan dengan candidate pattern generation saat ini. Metode FP-growth juga telah diimplementasikan pada versi terbaru sistem DBMiner dan dites di basis data industri besar, seperti basis data London Drugs, dengan performa yang memuaskan.

Saran yang dapat diberikan yakni jika basis data terlalu besar dan tidak realistis untuk dibangun main memory-based FP-tree, dapat dilakukan partisi dan setiap bagian partisi dibangun FP-tree. Alternatif lainnya yaitu dengan membangun sebuah disk-resident FP-tree.